

Universidad Carlos III de Madrid

Escuela Politécnica Superior

DISEÑO Y DESARROLLO DE UNA ESTACIÓN BASE FLUCTUANTE  
PARA MISIONES AÉREAS EN ENTORNOS FLUVIALES



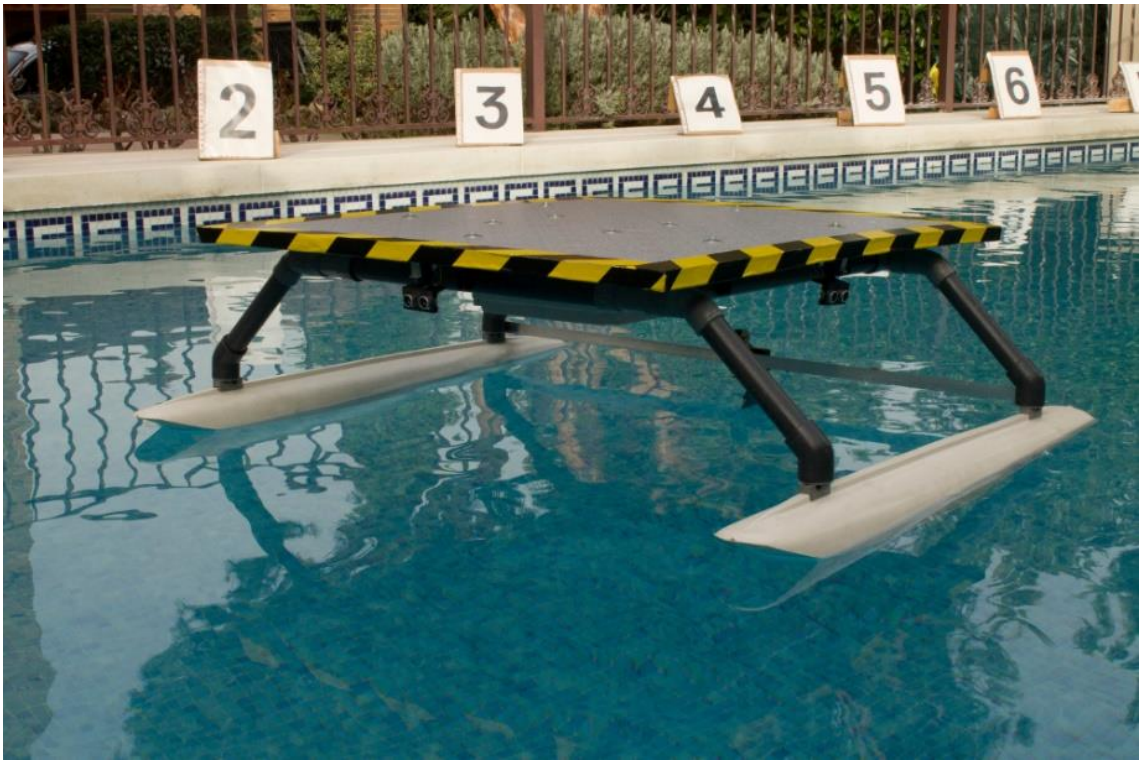
David Borreguero Díaz

Profesor: João Valente

Departamento de Ingeniería de Sistemas y Automática

Grado en Tecnologías Industriales

# STRIDER V1.0



## Índice

1. Introducción .....	1
1.1 Motivación .....	2
2.1 Objetivos .....	2
2. Estado del arte .....	3
3. Fases del proyecto.....	7
4. Diseño de la plataforma .....	8
4.1 Estudio de materiales.....	8
4.2 Diseño en CAD.....	12
5. Lista de materiales, presupuesto y pesos .....	15
6. Resultado del diseño aplicando los materiales elegidos.....	17
7. Cálculo de fuerzas y modelado dinámico.....	22
8. Hardware.....	33
8.1 Sistema de control.....	33
8.2 Sistema de orientación y aceleración .....	42
8.3 Esquema eléctrico .....	44
8.4 Componentes de navegación y transmisión .....	47
8.5 Otros elementos necesarios.....	50
9. Construcción del prototipo .....	53
10. Módulo de evasión de obstáculos.....	59
11. Descripción del funcionamiento .....	63
12. Pruebas y ensayos .....	67
13. Problemas y limitaciones encontrados .....	73
14. Posibles extras y mejoras futuras.....	74
15. Conclusiones.....	77
16. Bibliografía .....	78

## Tabla de ilustraciones

1. MARINER 600 (UST, 2015) .....	3
2. JUST READ THE INSTRUCTIONS (NASASPACEFLIGHT, 2016) .....	4
3. PROYECTO DE INVESTIGACIÓN DE LA FAU (GPSWORLD, 2015) .....	5
4. RIVERWATCH'S .....	6
5. PROCEDIMIENTO A SEGUIR .....	7
6. NICON DELTA (TAKEALOT, 2016) .....	8
7. HOVERCRAFT (RAIDENTECH, 2016) .....	8
8. TUBOS PVC 32MM (POOLARIA, 2016) .....	9
9. FLOTADORES PARA HIDROAVIONES RC DE ABS (HOBBYKING, 2015) .....	9
10. FLOTADORES PARA HIDROAVIONES RC DE POLIESPAN (HOBBYKING, 2015) .....	10
11. METACRILATO (METACRILAT, 2015) .....	10
12. POLICARBONATO CELULAR (LEROYMERLIN, 2016) .....	11
13. PLANCHA DE ESPUMA DE POLIURETANO (POLIURETANOS, 2015) .....	11
14. MEDIDAS DEL ARDRONE 2.0 (ARDRONE, 2015) .....	12
15. DISEÑO CAD DEL CONJUNTO .....	12
16. DISEÑO EXPLOSIONADO .....	13
17. PLANO VISTA SUPERIOR ACOTADO .....	13
18. PLANO VISTA INFERIOR ACOTADO .....	14
19. PLANCHA DE ESPUMA DE POLIURETANO .....	17
20. TUBOS PVC 32MM .....	17
21. ACCESORIOS PVC .....	17
22. UNIÓN PATÍN-ESTRUCTURA .....	18
23. PATINES ABS .....	18
24. ESTRUCTURA A FALTA DE LA BASE .....	19
25. UNIÓN ESTRUCTURA-PLANCHA .....	19
26. PLETINA ESTRUCTURAL DE ALUMINIO 20X20X1000 MM .....	20
27. CAJA DE ELECTRÓNICA ESTANCA .....	20
28. ESTRUCTURA FINAL .....	21
29. REPRESENTACIÓN DE FUERZAS .....	22
30. MEDIDAS PARA EL CÁLCULO DE ÁREAS .....	24
31. DESCOMPOSICIÓN DE LA VELOCIDAD .....	29
32. DISTRIBUCIÓN DE PRESIÓN .....	30
33. SHARP GP2Y0A21YK0F (WATTEROTT, 2009) .....	33
34. ALCANCE DEL SHARP GP2Y0A21YK0F (DATASHEET, 2006) .....	33
35. HC SR04 .....	34
36. TABLA DE CARACTERÍSTICAS HC SR04 (ZONAMAKER, 2016) .....	34
37. TEST DE ALCANCE (DATASHEET, 2010) .....	35
38. SERVO SG92R .....	35
39. SERVO WATERPROOF .....	36
40. L298N .....	37
41. MOTOR .....	38
42. MULTIPLEXOR DE PINES ARDUINO (SPARKFUN, 2016) .....	38
43. ARDUINO MEGA 2560 .....	39
44. PROGRAMA ARDUINO .....	39
45. MÓDULO BLUETOOTH HC 06 .....	40
46. APLICACIÓN (ARDUMOTIVE, 2013) .....	40
47. MÓDULOS RF (DX, 2016) .....	41
48. BATERÍA LRP .....	41
49. CARGADOR BATERÍA LRP .....	42

50. IMU GY-85 .....	42
51. SISTEMA DE CONTROL.....	44
52. SISTEMA DE BARRIDO.....	45
53. OTROS COMPONENTES .....	45
54. COMPONENTES DEL EJE DE TRANSMISIÓN .....	47
55. CARDÁN .....	48
56. REDUCTORA 3:1.....	48
57. HÉLICE PVC 40MM.....	49
58. TIMÓN .....	49
59. SOPORTE DEL SENSOR ULTRASONIDOS.....	50
60. DISEÑO 3D SOPORTE ULTRASONIDOS (THINGIVERSE, 2013).....	50
61. IMPRESIÓN 3D .....	50
62. CONJUNTO SOPORTE SERVO-HC04.....	51
63. INTERRUPTOR DE CORRIENTE .....	51
64. ZUMBADOR .....	51
65. LECTOR DE TARJETAS MICROSD .....	52
66. MONTAJE DEL MOTOR Y LA REDUCTORA .....	53
67. UNIÓN CARDÁN-EJE .....	53
68. CAJA DE CONEXIONES .....	54
69. COMPONENTES ADICIONALES.....	54
70. CLEMAS DE CONEXIÓN .....	54
71. INSTALACIÓN DEL SOPORTE DEL SERVO DE BARRIDO .....	55
72. INSTALACIÓN DEL SERVO DEL TIMÓN .....	55
73. VISTA DEL EJE.....	55
74. INSTALACIÓN DEL TIMÓN.....	56
75. ADAPTACIÓN DEL ZUMBADOR.....	56
76. INTERRUPTOR DE PUESTA EN MARCHA .....	57
77. ENCAPSULADO DEL ACELERÓMETRO .....	57
78. UBICACIÓN DEL ACELERÓMETRO .....	57
79. INSTALACIÓN ELECTRÓNICA.....	58
80. RESULTADO DE LA INSTALACIÓN .....	58
81. CAMPO DE ACCIÓN HORIZONTAL DE LOS SENSORES.....	59
82. TRIÁNGULO DE DISTANCIAS .....	59
83. CAMPO DE ACCIÓN VERTICAL DE LOS SENSORES .....	60
84. COMPORTAMIENTO REACTIVO .....	60
85. GIRO PROPORCIONAL DEL TIMÓN .....	61
86. REGULADOR PROPORCIONAL.....	61
87. ARDUMOTIVE. PANTALLA PRINCIPAL.....	63
88. ARDUMOTIVE. DISPOSITIVOS BLUETOOTH .....	63
89. ARDUMOTIVE. CONEXIÓN .....	64
90. ARDUMOTIVE. VELOCIDAD.....	64
91. FUNCIONES DE MEDIDA .....	65
92. FUNCIONES DIRECCIÓN Y MOVIMIENTO.....	65
93. FUNCIONES VARIAS .....	66
94. PRUEBA LÍNEA RECTA .....	68
95. GRÁFICO ORIENTACIÓN. LÍNEA RECTA .....	68
96. GRÁFICO VELOCIDAD. LÍNEA RECTA .....	68
97. PRUEBA U .....	69
98. GRÁFICO ORIENTACIÓN. U .....	69
99. GRÁFICO VELOCIDAD. U .....	69
100. PRUEBA EVASIÓN DE OBSTÁCULOS .....	70
101. GRÁFICA ORIENTACIÓN. EVASIÓN DE OBSTÁCULOS.....	70

102. GRÁFICA VELOCIDAD. EVASIÓN DE OBSTÁCULOS.....	70
103. GRÁFICAS DISTANCIA SENSORES Y TIMÓN. EVASIÓN DE OBSTÁCULOS .....	71
104. PLACA FOTOVOLTAICA (AMAZON, 2016) .....	74
105. ALCANCE DEL SHARP 1 (SCIELO, 2016) .....	75
106. ESTERA CON PUNTOS ANTIDESLIZANTES (MERCAMANIA, 2016) .....	75
107. COMUNICACIÓN DRON-BALSA.....	76
108. LED ROJO (STEREN, 2016).....	76

# 1. Introducción

Hoy en día, el uso de drones está cada vez más extendido. Podemos encontrar, por ejemplo, este tipo de vehículos aéreos en entornos militares ya que este es uno de sus usos más habituales, por lo que fue aquí donde se desarrollaron en un principio extendiéndose después a entornos civiles. Otra aplicación muy común de estos dispositivos es la de grabación, debido a su capacidad de elevación y su pequeño tamaño es capaz de realizar grabaciones con cierta perspectiva y muy cercanas consiguiendo imágenes muy difíciles de obtener de cualquier otra forma a nivel del suelo.

También son usados como dispositivos de rescate, ya que en caso de catástrofes naturales es posible que estos transporten botiquines, flotadores u otros objetos útiles que puedan salvar la vida a una persona, lo mismo se puede aplicar en playas para bañistas que sufran alguna lesión durante un baño o que se vean impedidos para volver a tierra. Además, facilita la búsqueda de personas ya que con la cámara que puede llevar incorporada unida a un software de reconocimiento es relativamente sencillo la localización de persona en medios adversos.

La NASA los usa para misiones de reconocimiento de fenómenos naturales, como fue el sobrevuelo del volcán Turrialba en Costa Rica en el que fueron capaces de tomar datos sobre la concentración de gases para realizar mediciones y predicciones durante una erupción de dicho volcán cuando se encontraba activo. También han sido usados por otras empresas en incendios y en defensa del medio ambiente, detectando comportamientos irregulares en las explotaciones de recursos.

Hay muchos otros usos que se están investigando los cuales serán muy frecuentes en un futuro cercano. Entre ellos se encuentra el reparto a domicilio, periodismo aéreo, monitorización de cultivos y búsquedas arqueológicas, vigilancia de actividades ilegales, manipulación de materiales nocivos, investigaciones biológicas, etc.

Debido al uso ya tan extendido de este tipo de dispositivos, este trabajo se centrará en esta área de investigación. En concreto, en los drones con misiones en los que el agua es un factor determinante. El objeto de este proyecto es la construcción de una base acuática de aterrizaje autónoma para UAV's (*Unmanned Aerial Vehicle*). Comprenderá desde el diseño de la base y la elección de los materiales, hasta la construcción del sistema electrónico y su correspondiente escritura de código. Una vez acabado dicho proceso se llevarán a cabo una serie de procedimientos y experimentos sobre el sistema para poner a prueba su funcionamiento.

## **1.1 Motivación**

Resulta un trabajo con la posibilidad de aportar algo muy significativo en su terreno, ya que actualmente es frecuente el uso de drones en el estudio y conservación del medio ambiente y no es extraño pensar en la necesidad de un elemento para situaciones de emergencia o abastecimiento en entornos en los que pueda no encontrarse el factor humano.

Este trabajo está centrado en el uso de drones en ambientes naturales. Es muy habitual hoy en día la recopilación de inteligencia para el control de ríos, riberas y cauces fluviales que permite conocer los estados de los mismos y sus cambios naturales mediante la monitorización aérea del terreno. Se usa tanto para el estudio de la fauna como de la flora, del estado de las aguas (contaminación y microbiología), incluso para la gestión de los medios de control de cauce tanto en la actividad diaria como en casos de riesgo de desbordamiento, inundaciones, etc.

En concreto, el objetivo de este trabajo es la construcción de una base de aterrizaje para entornos fluviales para vehículos aéreos no tripulados en este último tipo de usos. Se usará como elemento de apoyo ya que la base seguirá un cauce de agua para estar disponible en caso de que sea necesario su uso por parte del dron. Con esto podemos solucionar uno de los mayores problemas de este tipo de vehículos, la autonomía y/o las dificultades de vuelo en condiciones meteorológicas adversas. Para ello la base tendrá que tener un modo de navegación automático con sistema de evitación de obstáculos y un modo manual para poder controlar el movimiento del dispositivo de forma voluntaria por el operario.

## **2.1 Objetivos**

Podemos fijar los objetivos fundamentales de este trabajo en:

- Construcción de una estructura fuerte y resistente capaz de portar el sistema de control y un dron, al precio más reducido posible.
- Calcular las fuerzas y a consecuencia elegir los componentes necesarios para las condiciones planteadas en este proyecto.
- Dotar al sistema de un modo de navegación autónoma capaz de evitar los posibles obstáculos que pueda encontrarse en su trayectoria.
- Demostrar el funcionamiento y movilidad del dispositivo resultante.



## 2. Estado del arte

Antes de comenzar el proyecto establecer una base sobre los avances que ya existen relacionados con este trabajo.

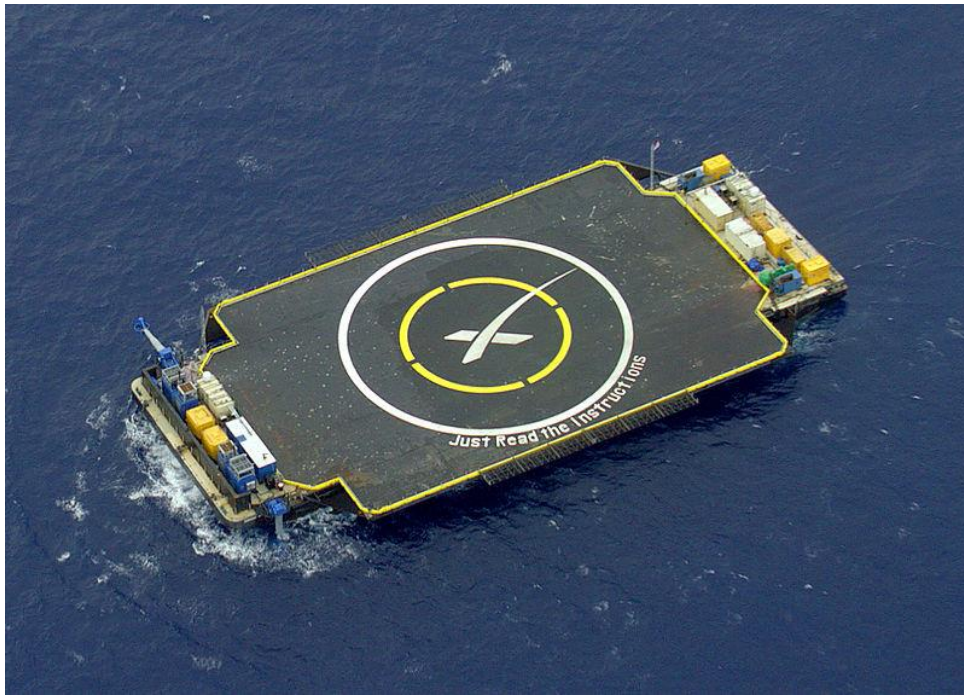
Un proyecto que puede estar relacionado con este trabajo es la adaptación de drones para su aterrizaje en medios acuáticos mediante la modificación de su estructura.



1. *Mariner 600* (UST, 2015)

En la imagen aparece fotografiado el *Mariner 600*. Es un dron adaptado para aterrizar en medios fluidos, también es una posible solución para casos de emergencia durante el vuelo. No obstante, existen algunas desventajas respecto a este proyecto. Principalmente, no se trata de un sistema universal aplicable a cualquier dron, ya que requiere la adaptación de cada uno de ellos en particular, por lo que hace necesaria una inversión por vehículo en vez de una para permitir el uso a modo general. Con este sistema cabe la posibilidad de realizar el aterrizaje, pero además de estar peligrosamente cercano al nivel del agua, en este caso no contaría con la posibilidad de la recarga de la batería ni el poder conducirlo hasta el punto deseado en el caso de que esta se haya agotado.

Ya desde 2014 se está trabajando en el diseño y construcción de una base de aterrizaje acuática para cohetes. Esta idea es algo muy reciente que apareció con la posibilidad de la realización de viajes espaciales a nivel civil, en los que para abaratar costes, se creó el concepto de vehículos espaciales reutilizables. La idea de esta investigación es la creación de un medio que permita usar un vehículo de transporte interplanetario más de una vez.



2. *Just Read the instructions* (Nasaspaceflight, 2016)

Esta base bautizada como “Just Read the instructions” posee unas dimensiones de 52 x 91 metros y es propulsada por 4 motores de 304 cv sumando un total de 1216cv. Esta base es capaz de estabilizarse incluso en condiciones acuáticas adversas y de mantener su posición con un error de 3 metros esperando el aterrizaje de un cohete. La embarcación es enviada a la posición de aterrizaje y está capacitada para que una vez realizado con éxito el proceso, volver autónomamente hasta un puerto preparado con unos anclajes especiales para su acople a tierra firma para concluir satisfactoriamente el viaje.

El primer intento de aterrizaje se produjo en Enero de 2015 por parte del *Falcon 9* durante la quinta misión de reabastecimiento de la estación aeroespacial internacional. El resultado fue fallido pues un error en el cálculo de la cantidad necesaria de fluido hidráulico, hizo que la plataforma y el cohete colisionaran provocando la destrucción de ambos.

Se han realizado dos versiones más posteriores a esta con resultados catastróficos en ambos, no obstante, las expectativas en este proyecto siguen siendo altas. Se espera que la reutilización de vehículos aéreos espaciales sea algo implantando en un futuro muy cercano (cfr. “SpaceX Augments and Upgrades Drone Ship Armada”, 2015)

Existe algo más cercano a este proyecto, aunque con diferentes propósitos. Se trata de un proyecto que realizó un equipo de investigación de la *Florida Atlantic University* (FAU), el cual desarrolló un vehículo autónomo acuático que puede asistir en inspecciones sobre el estado de puentes. Este proyecto recibió un premio de 187.000 dólares por parte del Departamento de transporte de Florida alegando que hace el trabajo de los inspectores de puentes más fácil y seguro. La función de este vehículo es

la revisión del estado de los puentes sobre todo, en los puntos que se tiene difícil acceso ya sea por la presencia de vegetación o por cualquier otro posible obstáculo del medio.



3. Proyecto de investigación de la FAU (GPSWorld, 2015)

Como sistema de navegación cuenta con un GPS y una brújula para el correcto posicionamiento y orientación durante las misiones de reconocimiento. Además, como futura mejora se contempla la posibilidad de añadirle un escáner acústico para la detección de desperfectos subacuáticos.

A día de hoy, existe una multitud de drones que interactúan con la naturaleza, entre los más famosos se encuentran el Airinov (Concebido para la mejora de las cosechas reduciendo el impacto ambiental), el Easyfly (para el estudio de aves pequeñas), el Global Hawk (para la predicción de huracanes), el Lifeseecker (localizador de personas en desastres naturales) y el Osprey (concebido para evitar la caza de ballenas).

Lo más parecido que podemos encontrar, pero a una escala totalmente distinta, es un portaaviones, ya que cuenta con un objetivo similar: aterrizar un vehículo aéreo en una superficie que flota en el agua. No obstante posee grandes diferencias, ya que la escala es diferente, además de la proporción, en el caso del portaaviones necesitamos una superficie de aterrizaje bastante alargada, debido a que la mayor parte de los aviones no cuentan con la posibilidad de realizar un aterrizaje de forma vertical al contrario que los VTOL (*Vertical Take-Off and Landing*).

En cuanto al uso de sensores ultrasonidos en medios acuáticos podemos encontrar a C. Detweiler que ha desarrollado ya una gran labor de investigación en este campo. En el artículo “Autonomous Aerial Water Sampling” (Detweiler *et. al.*, 2013) se habla sobre el uso de drones para tomar muestras de agua para el análisis de su composición en busca de contaminantes. En él, dicho autor desarrolla un sistema de recogida de muestras haciendo una comparación de las variaciones de medida al tomar los datos



con el sensor de presión y con el ultrasonido. Si bien, podría resultar útil una balsa de aterrizaje para este tipo de misiones, ya que las baterías del UAV impedirán, en algún momento, continuar el vuelo y podría incluso, servir como almacén temporal de los recipientes de muestras para poder coger otros vacíos pudiendo así reanudar la misión de muestreo.

Otro artículo en el que se trata un proyecto similar a este es: “An autonomous surface-aerial marsupial robotic team for riverine environmental monitoring: Benefiting from coordinated aerial, underwater, and surface level perception” (Pinto et al., 2014). Los investigadores han creado una base de aterrizaje con forma de catamarán, equipada con escáneres láser, sónar para la detección de objetos debajo del agua, GPS y una IMU. Posee además un sistema de visión multi-cámara, esta estructura está diseñada para contar con la posibilidad de viajar personas en ella, aunque no es necesario ya que cuenta modo de control autónomo. El proyecto también incluye un hexacóptero, dotado también con navegación autónoma soportado por un GPS y una IMU, todo ello en un entorno ROS.



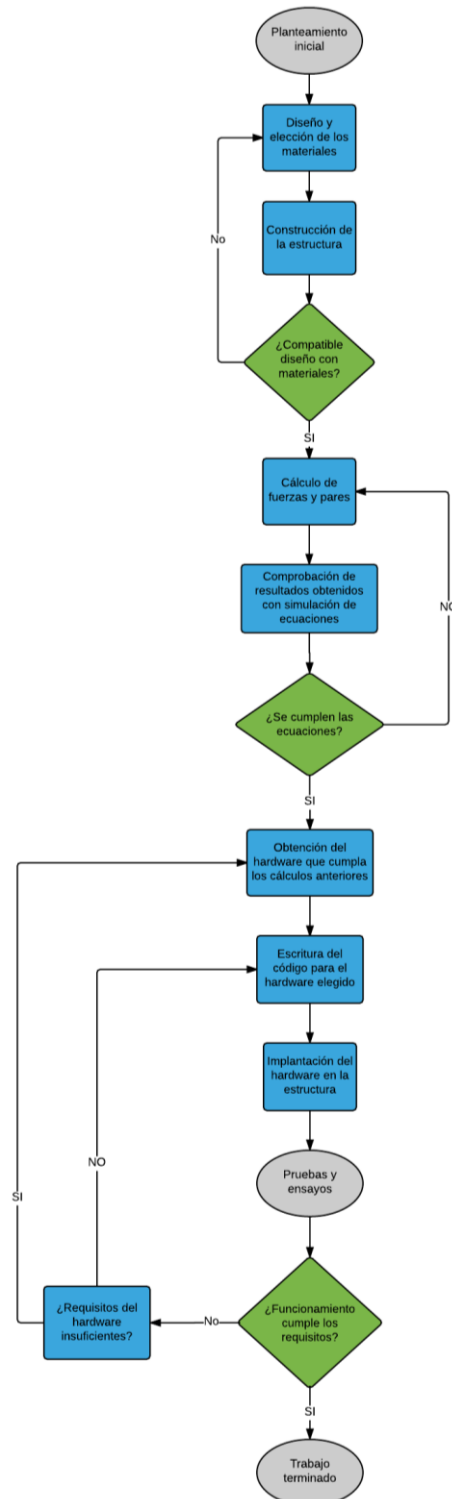
4. RIVERWATCH's

Ambos dispositivos cooperan para conseguir una misma misión, el UAV proporciona al catamarán una ampliación de su visión desde una perspectiva aérea, completando así el mapeo del territorio que este tenía. Con el reconocimiento del terreno de ambos se crean mapas en 3D modelando la forma geográfica del medio en el que se encuentran permitiendo así al sistema de navegación encontrar una ruta por la que evitar los obstáculos.

Aunque este último es muy parecido, no existe algo igual a este proyecto. Existen trabajos similares pero con diferentes funciones, objetivos y/o dimensiones, por lo que he usado una mezcla de las características más interesantes de estos que pueden servir para la misión de este proyecto.

### 3. Fases del proyecto

En este apartado, vemos en forma de diagrama de flujo, el procedimiento a seguir para el correcto desarrollo y construcción de este trabajo. Este comprende desde la elección de materiales hasta la construcción de la maqueta. Incluye los posibles bucles que surgirán en caso de que algo no funcione correctamente y haya que corregir algo.



5. Procedimiento a seguir

## 4. Diseño de la plataforma

### 4.1 Estudio de materiales

Desde un principio eran múltiples las opciones de diseño. En un principio este proyecto iba a tomar el fuselaje de una lancha de radiocontrol como superficie de flotabilidad. La imagen que viene a continuación fue durante un tiempo una propuesta de implantación.



6. *Nicon Delta (Takealot, 2016)*

Esta idea fue rápidamente desechada ya que estéticamente no era muy adecuado y además era difícilmente adaptable a la estructura que después los objetivos de este proyecto requerirían.

Otra de las ideas que vinieron a continuación era basarse en la estructura tipo *hovercraft*, en la que la superficie de aterrizaje estuviese asentada sobre una cámara de aire permitiendo así su flotabilidad en el agua.



7. *Hovercraft (Raidentech, 2016)*

Esta opción fue también desechada ya que al tener que utilizar una plataforma de grandes dimensiones debido al tamaño del dron, parecía una opción inviable en cuanto a su construcción.

Después de ir barajando las distintas posibilidades fue poco a poco acercándose a lo que sería el diseño final. Interesaba hacer una estructura en la que la base de aterrizaje estuviese a cierta altura de la superficie del agua para evitar problemas de contacto entre el agua y el dron y además, tendría que existir algún sitio en el que albergar todos los componentes necesarios para el funcionamiento de la base.

Después de continuar con la investigación llegué a la conclusión de que un catamarán era la forma más adecuada para basarse en este proyecto, debido a su morfología y distribución.

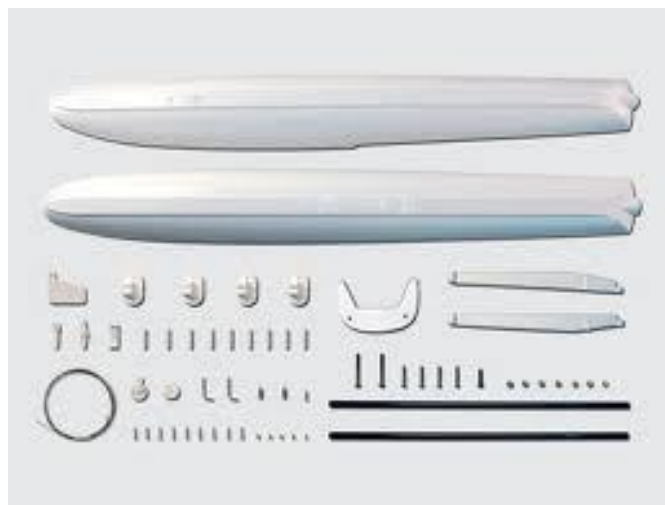
La opción más viable teniendo en cuenta la relación rigidez/precio era la de realizar una estructura con tubos de PVC en la que apoyar una plancha de algún material que sirviera como zona de aterrizaje.

Después de llegar a este punto ya solo falta elegir el material para terminar el planteamiento de la estructura. Esta estaba ya decidida: tubos de PVC, debido a su peso ligero, precio económico y facilidad de manipulación. Pero aún quedaban dos materiales que elegir: la base de aterrizaje y el material que hiciese flotar al conjunto.



8. Tubos PVC 32mm (Poolaria, 2016)

Una vez elegida la estructura es toca pasar a la búsqueda del material que sirva como elemento de flotabilidad a la estructura. Los aviones radiocontrol cuentan con la posibilidad de añadir un accesorio adicional, se trata de patines de aterrizajes acuáticos. Después de un largo proceso de selección estos fueron los elegidos:



9. Flotadores para hidroaviones RC de ABS (Hobbyking, 2015)

Por estar fabricados en ABS y debido a sus grandes dimensiones (91cm) los hacían ideales para este proyecto, ya que la opción alternativa era usar los conocidos “churros

de piscina” de espuma de polietileno, pero estos resultan ser menos fiables y menos estéticos.

También, existía la posibilidad de usar este mismo tipo de accesorio pero fabricado en Elapor (Poliestireno expandido, “poliespan”):

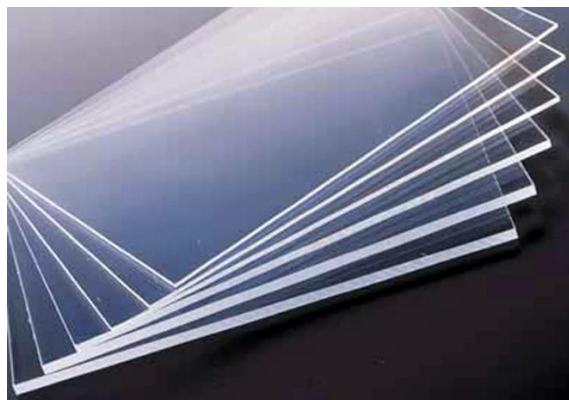


10. Flotadores para hidroaviones RC de poliespan (Hobbyking, 2015)

A pesar de ser algo más económicos fueron descartados debido a que el peso que soportan es menor, además de tener menos resistencia. Por otra parte, al tratarse de un diseño para cauces de agua podríamos encontrarnos con diferentes tipos de obstáculos y corrientes, lo cual nos obliga a usar una estructura más robusta y resistente.

Una vez planteada la mayor parte de la estructura ya solo queda decidir de qué material estaría compuesta la base de aterrizaje para el dron. Aquí fue donde el proceso de búsqueda fue más largo, ya que era necesario un material que fuese rígido, ligero y resistente al agua. Por este último motivo fue descartada la madera, ya que requiere tratamientos específicos para su resistencia acuática y eso conlleva el hecho de ser más propensa a deterioros. El objetivo es hacer una estructura capaz de resistir agentes atmosféricos y un cierto grado de condiciones adversas, por lo que enseguida finalmente un material polimérico era la mejor opción.

Una de las primeras opciones barajadas una vez llegado a este punto fue el metacrilato.

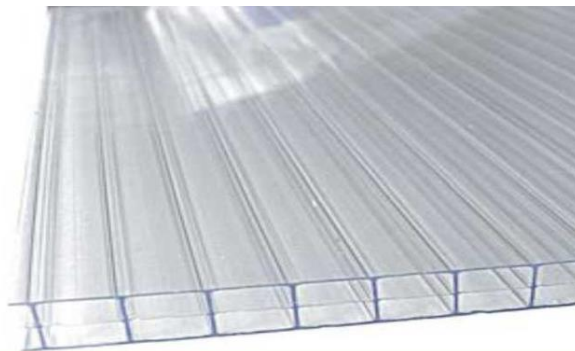


11. Metacrilato (Metacrilat, 2015)



Rápidamente fue descartado debido a que para obtener la resistencia requerida necesita de un gran espesor, lo que conlleva un exceso de peso debido a la alta densidad de este material además de su alto precio.

Un material más ligero que, debido a su estructura tipo “sándwich” con celdas en el interior, consigue una buena resistencia con un peso reducido es el policarbonato celular:



*12. Policarbonato Celular (Leroymerlin, 2016)*

Esta opción fue desechada debido también a su elevado precio, ya que también es objetivo de este proyecto el realizarlo al menor coste posible, y no era una buena decisión el destinar la mayor parte del presupuesto a esta zona.

Una opción mucho más adecuada a este proyecto es una plancha fabricada con espuma de poliuretano encerrada entre dos finas láminas de aluminio para evitar el deterioro de la espuma. A pesar de sus 2cm de espesor, se trata de un material muy ligero y muy resistente, por lo que este es el material elegido como plataforma de aterrizaje.



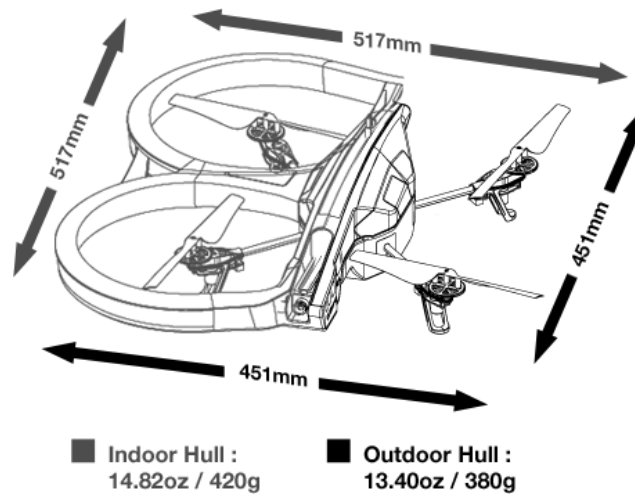
*13. Plancha de espuma de poliuretano (Poliuretanos, 2015)*

Para los anclajes entre las piezas de PVC y la plancha, ha sido necesaria la utilización de varios componentes. Los tubos han sido unidos a la plancha con unas pinzas para tubos de PVC. Dichas pinzas van sujetas por un tornillo pasante a la plancha de poliuretano, en la cual se ha colocado un conjunto tuerca-arandela para sujetar la pinza al otro extremo, encajando los tubos de la estructura directamente sobre ellas. Este elemento será descrito más adelante en este escrito (Apartado 6).

Una vez escogidos todos los materiales, ya solo quedaba llevarlo a la práctica y construirla de forma real, pero para ello antes es necesaria la realización de un diseño en CAD para optimizar todos los detalles y evitar el mayor número de errores posibles.

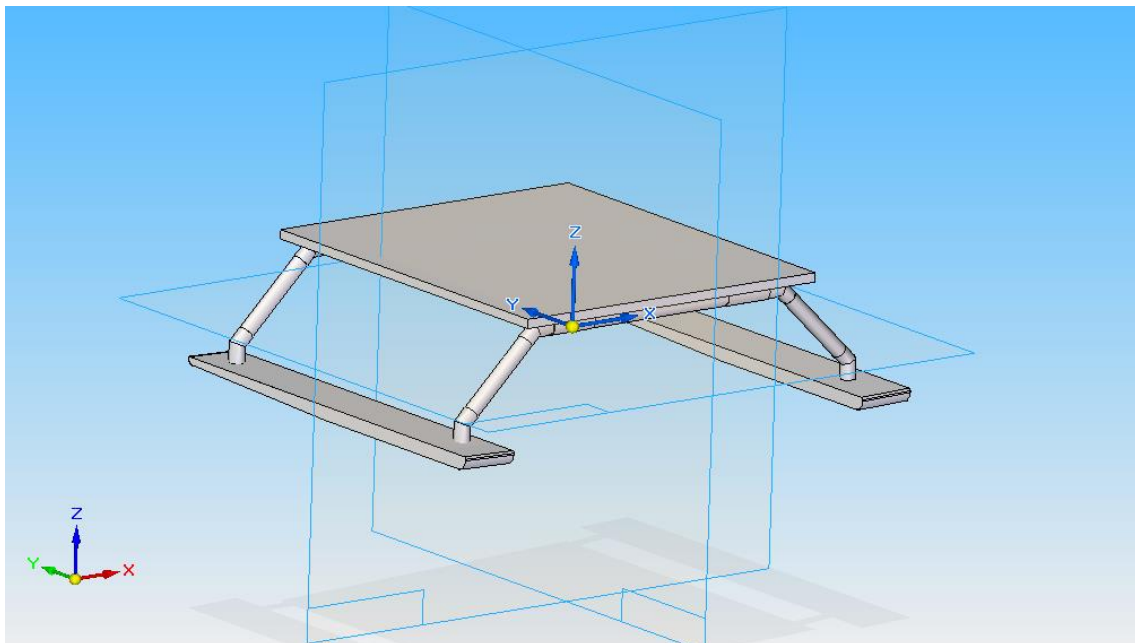
## 4.2 Diseño en CAD

Finalmente, una vez elegidos los materiales, fijadas las dimensiones y teniendo en cuenta las medidas del dron para el que está enfocado el diseño (figura de abajo) de esta base de aterrizaje:



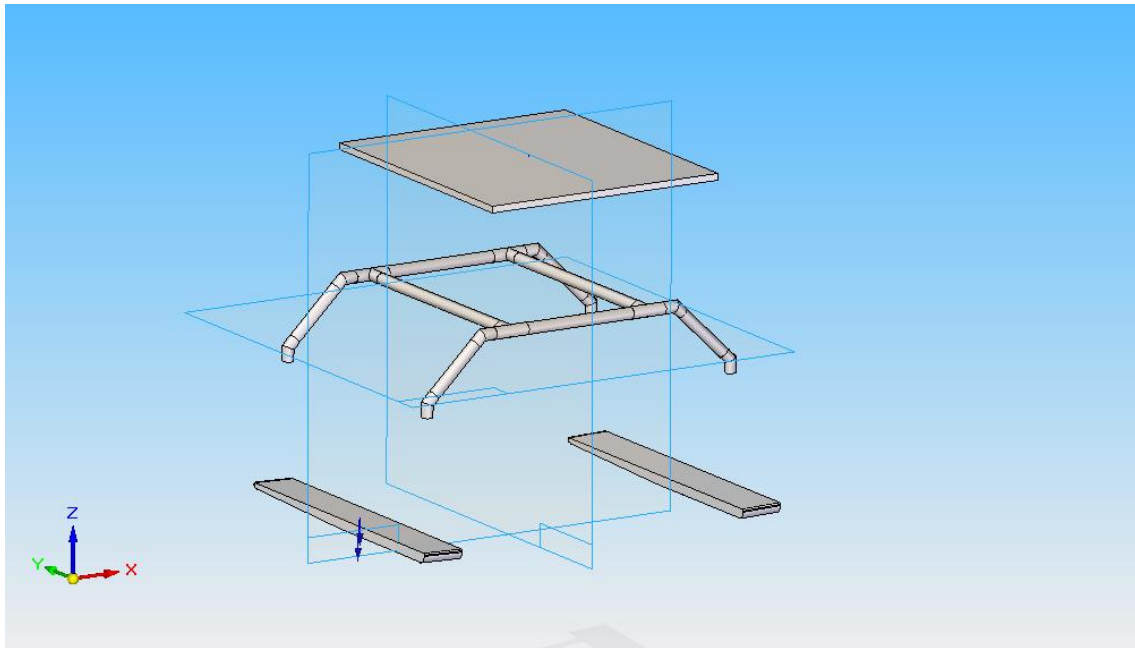
14. Medidas del Ardrone 2.0 (Ardrone, 2015)

Este fue el resultado:



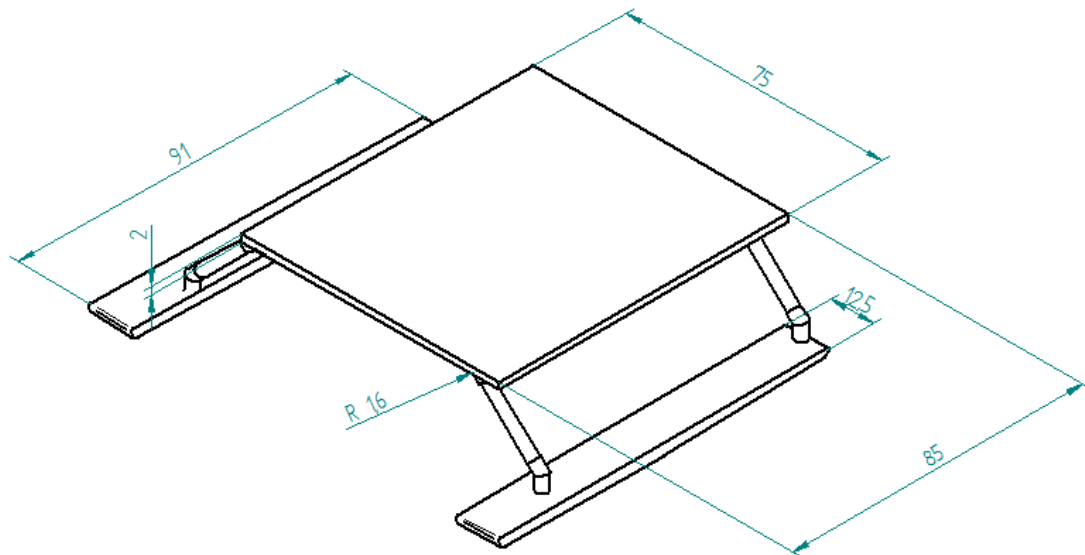
15. Diseño CAD del conjunto

Con un poco más de detalle podemos observar el diseño explosionado del conjunto:

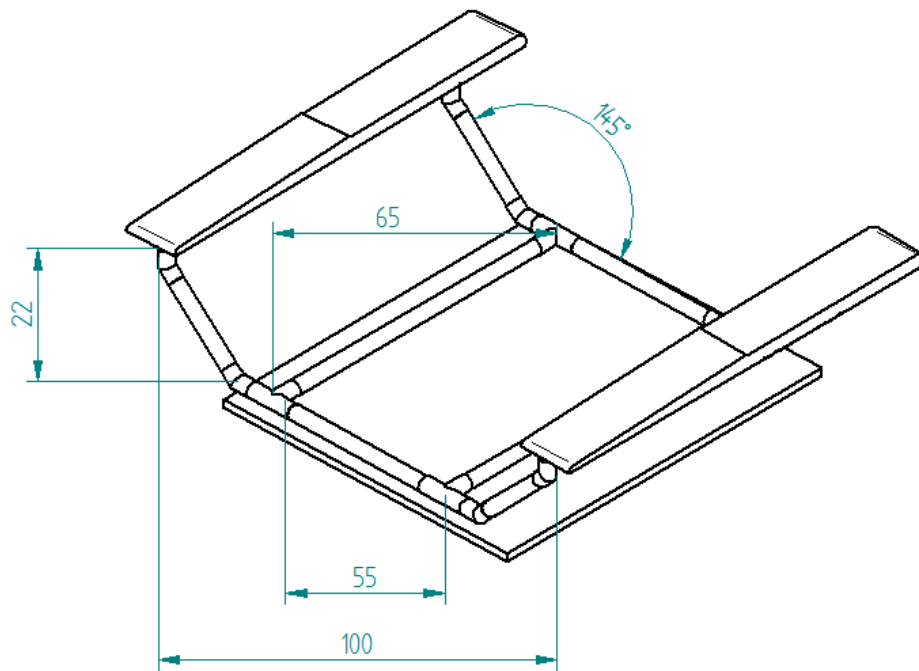


16. Diseño explosionado

En las siguientes imágenes se pueden apreciar las medidas del diseño detalladas:



17. Plano vista superior acotado



18. Plano vista inferior acotado

El ángulo formado entre el plano horizontal de la base y las patas es de  $145^\circ$ . Esto se ha conseguido debido a que se han usado unas T's de PVC para formar el marco que soporta la superficie de aterrizaje, y unido a ella un codo de  $45^\circ$ , de forma que nos da la inclinación suficiente para llegar hasta el patín. Al final de la pata se encuentra otro codo de  $45^\circ$  de forma que compense la inclinación del primero para acabar en perpendicular respecto a la base. De esta forma se consigue que la pata acabe de forma paralela al patín para poder así facilitar su unión. El marco de PVC queda centrado en el centro de la plancha, resultando equidistantes las distancias salientes de las patas respecto al tablero.

## 5. Lista de materiales, presupuesto y pesos

En este apartado veremos un resumen de los materiales usados en este proyecto, en el cual, además de ver la descripción de los elementos, podremos ver su cantidad, precio y peso, tanto unitario como en la cantidad total usada.

Por un lado, tenemos la parte estructural. Aquí se encontraran todos los materiales usados para la geometría de la balsa, además de todo lo necesario para su adhesión y refuerzo. En la siguiente parte, la motriz, tenemos todos los elementos usados para permitir el desplazamiento y direccionamiento del sistema. Y por último, tenemos la parte electrónica, la cual albergará todo lo necesario para el control y funcionamiento del proyecto.

En esta tabla vemos todos los componentes usados, sin embargo, hay uno en concreto que no cuenta para el cómputo global del peso del aparato. El cargador de baterías no habrá que tenerlo en cuenta en el sumatorio de peso a la hora de calcular las fuerzas del sistema, ya que es un componente que no viajará a bordo durante el trayecto.

En la página siguiente por tanto, se muestra la tabla de materiales.

Material	Cantidad	Precio / u	Precio Total	Peso / u (gr)	Peso (gr)
<b>ESTRUCTURA</b>			<b>102,57 €</b>	<b>3322,5</b>	
Patines 91 cm	1	52,41 €	52,41 €	652	652
Tubo rígido PVC 32mm	3 metros	1,23 €	3,70 €	994	994
Codo PVC 45º 32mm	8	0,52 €	4,16 €	32	280
T PVC 32mm	4	1,17 €	4,68 €	62	248
Tapa PVC 32mm	4	0,50 €	1,70 €	8	32
Pinza tipo clip PVC 32mm	6	0,74 €	4,44 €	24	144
Plancha espuma poliuretano 75x85cm	1	7,00 €	7,00 €	420	420
Pegamento PVC	1	1,28 €	1,28 €	125	62,5
Pletina estructural aluminio Perfil 20x20x1,5 mm -- 1 metro	1	2,20 €	2,20 €	142	142
Caja de plastico electricidad estanca 240x180x90mm	1	6,00 €	6,00 €	300	300
Conjunto soporte servo y HC04	4	3,75 €	15,00 €	12	48
<b>PARTE MOTRIZ</b>			<b>62,79 €</b>	<b>302,5</b>	
L298N Motor Drive Shield	1	8,80 €	8,80 €	30	30
Motor 7.2V 6A	1	15,90 €	15,90 €	73	73
Helice 40mm (3 hojas)	1	3,50 €	3,50 €	2	2
Tubo cobre 6mm ext - 4mm int	1	4,87 €	4,87 €	66	66
Varilla acero 4mm	1	4,87 €	4,87 €	66	66
Cardan 4mm	1	5,95 €	5,95 €	6	6
Timón	1	5,63 €	5,63 €	39,5	39,5
Reductora Graupner 3:1	1	13,27 €	13,27 €	20	20
<b>COMPONENTES ELECTRÓNICOS</b>			<b>180,30 €</b>	<b>685</b>	
Batería LRP 4000 mAh 7.2V 6 cell	1	31,00 €	31,00 €	390	390
Mazo de cable	10 M	3,00 €	3,00 €	13	130
Interruptor	1	1,50 €	1,50 €	6	6
Conector alimentacion arduino	1	1,00 €	1,00 €	3	3
Módulo microSD Arduino	1	7,50 €	7,50 €	5	5
Cargador 20W 2 A	1	15,00 €	15,00 €	188*	188*
Servos TowerPro SG92R	4	6,50 €	26,00 €	9	36
Servo waterproof para timón	1	11,90 €	11,90 €	38	38
Arduino Mega	1	31,00 €	31,00 €	25	25
IMU GY85	1	18,00 €	18,00 €	9	9
HC SR 04	4	5,00 €	20,00 €	10	40
HC 06	1	14,40 €	14,40 €	3	3
<b>TOTAL PROYECTO</b>			<b>345,66 €</b>	<b>4310</b>	

\*No computa en el peso de la plataforma

## 6. Resultado del diseño aplicando los materiales elegidos

Aquí vemos los distintos elementos que se han usado para formar la estructura, la superficie de aterrizaje:



19. Plancha de espuma de poliuretano

Los tubos usados para hacer la estructura que se encarga de la sustentación:



20. Tubos PVC 32mm

Y los distintos accesorios usados para su forma y unión:



21. Accesorios PVC



1. *Tes PVC 32mm*
2. *Codo PVC 45º 32mm*
3. *Codo PVC 45º 32mm*
4. *Tapa final de pieza 32mm*
5. *Pegamento de acción lenta para PVC*
6. *Pieza de unión al carril del patín*
7. *Pinza tipo clip PVC (con tornillo, tuerca y arandela)*

Los patines han sido unidos a la estructura a través de un pequeño conjunto que he creado, usando un codo de PVC de 45 grados (es necesario compensar el ángulo con el que viene la pata gracias al codo de 45 de arriba con este para que la zona de unión al patín acabe de forma paralela al suelo y facilitar su acople), una tapa de final de pieza y una de las piezas suministradas con los patines para facilitar la unión del codo al carril de estos.



*22. Unión patín-estructura*

Una vez conseguidos estos sistemas de unión, podemos unir al resto de la estructura a los patines elegidos:



*23. Patines ABS*



Después de hacer este acople cuatro veces (una para cada pata), cortando los tubos y usando los distintos accesorios que componen la estructura, este es el resultado:



*24. Estructura a falta de la base*

Después, con el uso de las pinzas de PVC unido al conjunto Tornillo-Tuerca-Arandela creamos el sistema de unión de la estructura a la plancha, llevando seis de estos, dos en los tubos más largos y uno solo en los lados cortos.



*25. Unión estructura-plancha*

Uniendo la base de aterrizaje al resto de la estructura obtenemos la forma final. Una vez acabada la estructura tuve que hacer un par de añadidos para necesidades futuras. Una de las modificaciones fue añadir un ángulo de pletina de aluminio para tener un soporte para el timón y su servo correspondiente. Esta pletina uniría las dos patas traseras a la altura superior del patín, obteniendo así un punto de apoyo cercano al agua.



*26. Pletina estructural de aluminio 20x20x1000 mm*

Otro añadido que hubo que realizar es el acople de una caja de electrónica estanca, en ella se ubicarían todos los componentes electrónicos necesarios tales como Arduino, la batería, el motor, etc. Se trata de una caja de plástico hermética con agujeros sellados por tapones de plástico, los cuales nos permitirán comunicar todos los componentes con el exterior.



*27. Caja de electrónica estanca*

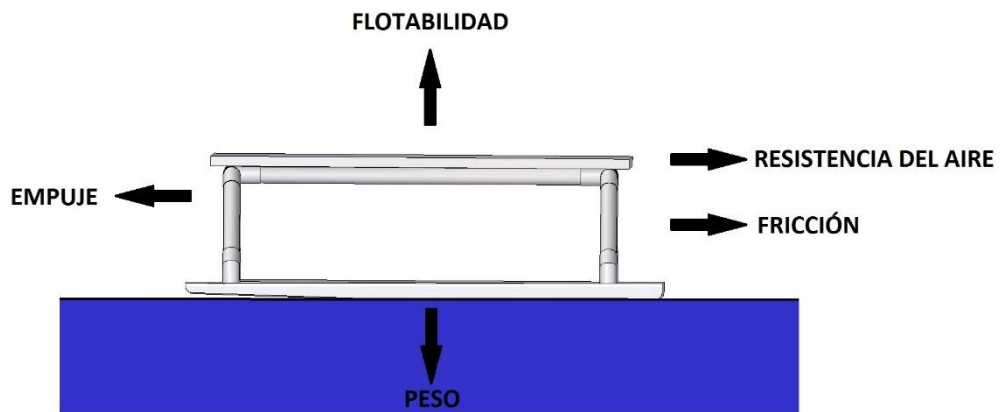
Una vez añadidas estas modificaciones finales este es el resultado obtenido:



*28. Estructura final*

## 7. Cálculo de fuerzas y modelado dinámico

Debemos hacer un estudio de todas las fuerzas que afectan a la estructura en su navegación. Aquí abajo vemos un ejemplo de las principales fuerzas que tenemos en un cuerpo en movimiento sobre un curso de agua:



29. Representación de fuerzas

Tenemos la fuerza de flotabilidad ( $f$ ) que compensa el peso ( $W$ ) de la embarcación. Sabiendo que el sumatorio de fuerzas verticales tiene que ser igual a cero para mantener el sistema en equilibrio, tomando el eje positivo como ascendente, obtenemos que:

$$\sum F_v = 0$$

$$f - W = 0$$

$$f = W$$

Expresando el peso en función de la masa:

$$f = m * g$$

Siendo:

$$f = \text{Flotabilidad (N)}$$

$$m = \text{Masa total del cuerpo (Kg)}$$

$$g = \text{Gravedad de la tierra (m/s}^2\text{)}$$

Tomando la gravedad como  $9,81 \text{ m/s}^2$  y sabiendo que la masa de la estructura, según los cálculos estimados en la tabla de materiales (Apartado 5) es de  $4,31 \text{ Kg}$ , podremos proceder a realizar los cálculos. A esto se le debe restar la masa de los patines, ya que para hacer el estudio de flotabilidad necesitamos obtener la masa que deben soportar estos, por lo que el dato a usar para realizar los cálculos será:

$$m_{\text{cálculos}} = m_{\text{Estructura}} - m_{\text{Patines}} = 4,31 \text{ Kg} - 0,652 \text{ Kg} = 3,658 \text{ Kg}$$

Para mayor comodidad en los cálculos tomaremos el dato aproximado de 3,66 Kg. Por lo tanto obtenemos el peso:

$$W = 3,66 \text{ Kg} * 9,81 \text{ m/s}^2 = 35,9 \text{ N}$$

La flotabilidad va a ser igual al peso:

$$W = f = 35,9 \text{ N}$$

Según los datos obtenidos por el fabricante los patines son capaces de soportar hasta 7 Kg (68,6 N), si incluimos el peso del dron (420 gramos (Parrot, 2016)) podemos concretar que:

$$W_{dron} = 0,42 \text{ Kg} * 9,81 \text{ m/s}^2 = 4,12 \text{ N}$$

El peso total a soportar será:

$$P_{Total} = P_{Estructura \text{ sin patines}} + P_{dron}$$

$$P_{Total} = 35,9 \text{ N} + 4,12 \text{ N} = 40,02 \text{ N}$$

Y como

$$40,02 \text{ N} < 68,6 \text{ N}$$

Podemos concluir que,

$$P_{Total \text{ a soportar}} < P_{max. \text{ admitido}}$$

Por lo que no tendremos ningún riesgo de hundimiento, ya que tenemos un gran margen hasta el peso máximo soportado. Podemos calcular esta diferencia de peso como el coeficiente de seguridad:

$$CS = \frac{P_{max. \text{ admitido}}}{P_{Total \text{ a soportar}}} = \frac{68,6}{40,02} = 1,71$$

Como vemos, podemos incrementar hasta en un 171% del peso hasta llegar al límite máximo soportado.

## Motor

Después de llegar a este punto, ya solo nos quedará calcular la fuerza total que el motor ha de vencer, que será igual a la suma de la resistencia del aire (Air resistance) más la de fricción del agua (Friction). Venciendo estas fuerzas conseguiremos el desplazamiento de la unidad por el agua.

$$F_{empuje} = F_{Resistencia \text{ Aire}} + F_{Fricción}$$

Lo que necesitamos es hallar la fuerza de arrastre para después buscar un motor que nos proporcione el impulso adecuado. Para ello, se deben fijar algunos valores antes de comenzar los cálculos, como por ejemplo fijamos la velocidad máxima de la plataforma en **5 m/s**.

Otros datos que necesitaremos serán las propiedades de los fluidos implicados. Para obtener dichos datos supondremos para todos los cálculos una temperatura media estándar de 20°C (293 K):

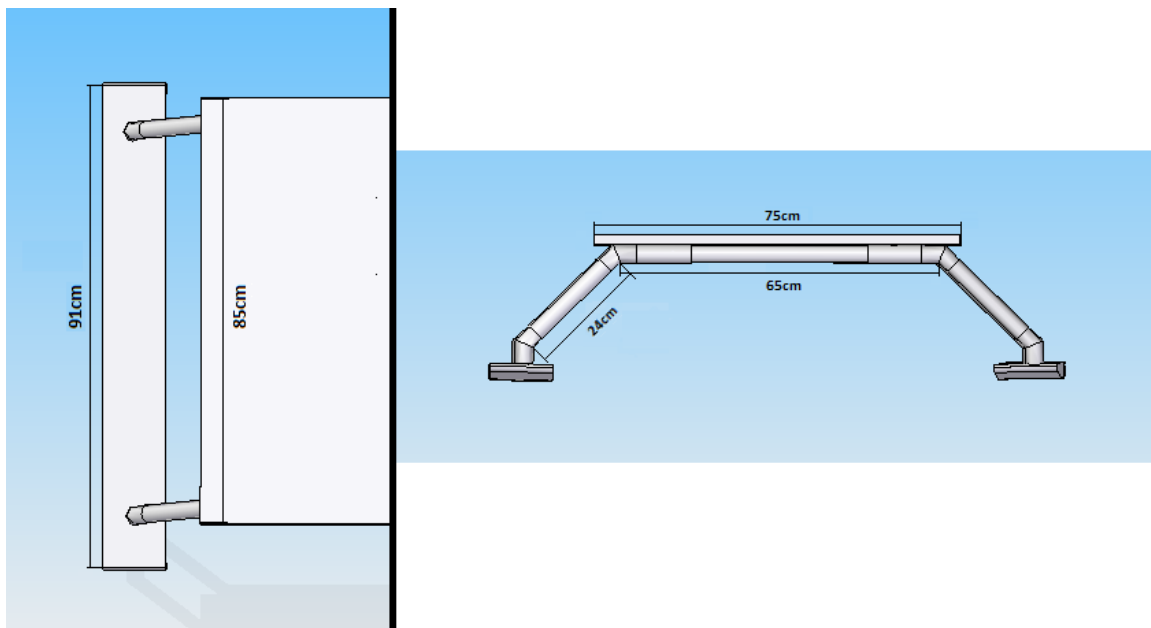
Propiedades del aire ( T=293 K)	
Densidad ( $Kg/m^3$ )	1,18
Viscosidad dinámica ( $Kg/m * s$ )	$1,8 * 10^{-5}$

Propiedades del agua ( T=293 K)	
Densidad ( $Kg/m^3$ )	998
Viscosidad dinámica ( $Kg/m * s$ )	$1,02 * 10^{-3}$

\*Datos obtenidos de *Fundamentos de transferencia de Calor*, Incropera (Véase bibliografía)

En realidad nos encontramos con varias fuerzas: la provocada por la resistencia de la plancha, la provocada por la estructura y la de los patines.

Nos hacen falta más datos para calcular las fuerzas existentes, los cuales recopilaremos paso por paso. Basando los cálculos en las medidas de las piezas reales obtenidas en el siguiente plano, se calculan primero todas las áreas que serán necesarias en los cálculos posteriores:



30. Medidas para el cálculo de áreas



$$A_{FrontalPlancha} = 0,75 \text{ m} \times 0,02 \text{ m} = 0,015 \text{ m}^2$$

$$A_{SuperficiePlancha} = 0,85 \text{ m} \times 0,75 \text{ m} = 0,64 \text{ m}^2$$

$$A_{FrontalTubos} = (0,65 \text{ m} + 2 \times 0,24 \text{ m}) \times 0,032 \text{ m} = 0,036 \text{ m}^2$$

$$A_{SuperficiePatín} = 0,91 \text{ m} \times 0,125 \text{ m} = 0,0114 \text{ m}^2$$

Se desprecia el área frontal del patín debido a que es mucho menor a las demás áreas consideradas.

Después tenemos que calcular qué tipo de régimen llevarán los fluidos. En el caso de la plancha será únicamente el determinado por el aire, mientras que en el caso de los patines será por un lado el del agua y por otro lado el del aire.

Comenzamos por calcular el de la plancha por ser el más sencillo. El método para ello es calcular el número de Reynolds y ver el tipo de flujo, esto lo sabremos viendo si:

$$Re < 5 \times 10^5 = \text{Flujo laminar}$$

O

$$5 \times 10^5 < Re < 10^7 = \text{Flujo turbulento}$$

Para después calcular su correspondiente coeficiente de fricción:

$$C_f = 1,33 \times Re^{-1/2} = \text{Flujo laminar}$$

O

$$C_f = 0,074 \times Re^{-\frac{1}{5}} = \text{Flujo turbulento}$$

Para calcularlo usaremos la expresión:

$$Re = \frac{\rho \times u \times L}{\mu}$$

Donde:

$$Re = \text{Número de Reynolds}$$

$$\rho = \text{Densidad del fluido (Kg/m}^3\text{)}$$

$$u = \text{Velocidad relativa del objeto (m/s)}$$

$$L = \text{Longitud característica (m)}$$

$$\mu = \text{Viscosidad dinámica del fluido (Kg/m} \times \text{s)}$$

Contamos con que el flujo va a ser paralelo al lado largo de la placa por lo que será esta medida la longitud característica a tener en cuenta para los cálculos. Tomando este dato y los anteriores tenemos que:

$$Re = \frac{1,18 \text{ Kg/m}^3 * 5 \text{ m/s} * 0,85 \text{ m}}{1,8 * 10^{-5}} = 2,79 * 10^5$$

Y como:

$$2,79 * 10^5 < 5 * 10^5$$

Interpretando el resultado vemos que obtenemos un flujo laminar. Sabiendo esto podemos usar la fórmula correcta para calcular el coeficiente de resistencia  $C_f$  para posteriormente calcular la fuerza. Para calcular este coeficiente usaremos la del cálculo de coeficiente de resistencia medio correspondiente a flujo en placa plana en régimen laminar:

$$C_f = 1,33 * Re^{-1/2}$$

Sustituyendo los datos obtendremos que:

$$C_{fPlancha} = 1,33 * (2,79 * 10^5)^{-\frac{1}{2}} = 0,0025$$

Como tenemos en realidad dos caras por las que se ofrecerá una resistencia, tendremos que multiplicar un factor de 2 el resultado anterior para obtener un resultado más real, de esta forma obtenemos el coeficiente de arrastre ( $C_D$ ) que será con el que calculemos la fuerza total a vencer:

$$C_{DPlancha} = 0,0025 * 2 = 0,005$$

Ahora pasaremos a calcular el coeficiente de resistencia de los patines. Para facilitar los cálculos asumiremos que los patines son rectangulares. Primero procedemos a calcular el Reynolds por el lado del aire:

$$Re_{patin-aire} = \frac{1,18 \text{ Kg/m}^3 * 5 \text{ m/s} * 0,91 \text{ m}}{1,8 * 10^{-5}} = 2,98 * 10^5$$

Y por otro lado el del agua:

$$Re_{patin-agua} = \frac{998 \text{ Kg/m}^3 * 5 \text{ m/s} * 0,91 \text{ m}}{1,02 * 10^{-3}} = 4,45 * 10^6$$

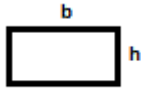
Por los márgenes que hemos visto anteriormente, ambos Reynolds nos indican que estamos en régimen turbulento por lo que calcularemos el coeficiente con la fórmula anterior:

$$C_{fAire} = 1,33 * (2,98 * 10^5)^{-\frac{1}{2}} = 0,0024$$

$$C_{fAgua} = 0,074 * (4,45 * 10^6)^{-\frac{1}{5}} = 0,0035$$

Por último, nos quedará calcular el  $C_f$  provocado por la estructura. Si la vemos de frente en realidad estamos viendo un rectángulo por lo que podemos usar la siguiente imagen, extraída de apuntes sobre resistencia aerodinámica, para aproximar su coeficiente.



<b>Placa plana rectangular</b> 	<b><math>b/h</math></b>	<b><math>C_f</math></b>
	1	1.18
	5	1.2
	10	1.3
	20	1.5
	$\infty$	2.0

\*Datos obtenidos de *Fundamentos de transferencia de Calor*, Incropera (Véase bibliografía)

Si tomamos  $b=1,13\text{m}$  (la suma de la longitud de los tubos frontales y de las dos patas) y  $h=0,032\text{m}$  (el diámetro del tubo):

$$\frac{b}{h} = \frac{1,13}{0,032} = 35,31$$

Según la representación anterior, podemos aproximar el  $C_f$  de la estructura al caso más extremo que es donde nos aseguramos que menos error vamos a cometer.

$$C_{fEstructura} = 2$$

Podemos usar el mismo coeficiente para el frontal de la plancha, que será necesario para calcular la resistencia del aire sobre la superficie perpendicular al flujo. Por lo que podemos deducir:

$$C_{fFrontalPlancha} = 2$$

Debido a que:

$$\frac{b}{h} = \frac{0,75}{0,02} = 37,5 > 20$$

Una vez obtenidos los 5 coeficientes, podemos calcular sus respectivas fuerzas que afectan en nuestra estructura a través de la fórmula:

$$F_f = \frac{1}{2} * \rho * u^2 * C_f * A$$

Donde:

$$A = \text{Area de contacto (m}^2\text{)}$$

$$C_f = \text{Coeficiente de fricción}$$

$$F_f = \text{Fuerza de arrastre (N)}$$

Por lo que podemos obtener:

$$F_{fPlancha} = \frac{1}{2} * 1,18 \text{ Kg/m}^3 * (5 \text{ m/s})^2 * 0,005 * 0,64 \text{ m}^2 = 0,047 \text{ N}$$

$$F_{fFrontalPlancha} = \frac{1}{2} * 1,18 \text{ Kg/m}^3 * (5 \text{ m/s})^2 * 2 * 0,015 \text{ m}^2 = 0,44 \text{ N}$$

$$F_{fPatín-Aire} = \frac{1}{2} * 1,18 \text{ Kg/m}^3 * (5 \text{ m/s})^2 * 0,0024 * 0,0114 \text{ m}^2 = 0,4 * 10^{-3} \text{ N}$$

$$F_{fPatín-Agua} = \frac{1}{2} * 998 \text{ Kg/m}^3 * (5 \text{ m/s})^2 * 0,0035 * 0,0114 \text{ m}^2 = 0,5 \text{ N}$$

$$F_{fEstructura} = \frac{1}{2} * 1,18 \text{ Kg/m}^3 * (5 \text{ m/s})^2 * 2 * 0,036 \text{ m}^2 = 1,06 \text{ N}$$

Por lo que de la suma de las 5 obtenemos la fuerza total:

$$F_T = F_{fPlancha} + F_{fFrontalPlancha} + 2 * F_{fPatín-Aire} + 2 * F_{fPatín-Agua} + F_{fEstructura}$$

$$F_T = 0,047 \text{ N} + 0,44 \text{ N} + 2 * 0,4 * 10^{-3} \text{ N} + 2 * 0,5 \text{ N} + 1,06 \text{ N} = 2,55 \text{ N}$$

Una vez que tenemos la fuerza total la convertiremos en potencia mediante la siguiente expresión:

$$P_T = F_T * u$$

$$P_T = 2,55 \text{ N} * 5 \text{ m/s} = 12,75 \text{ W}$$

Si a esto le añadimos el dato del rendimiento del 75% típico de los motores eléctricos tenemos que la potencia real es:

$$P_R = \frac{P_T}{0,75} = \frac{12,75 \text{ W}}{0,75} = 17 \text{ W}$$

Con esto obtenemos la potencia que nos es necesaria en el motor para impulsar la base a la que habrá que dejar un margen de potencia para las pérdidas de energía producidas por fricciones y calor además de la pérdida en el eje de transmisión.

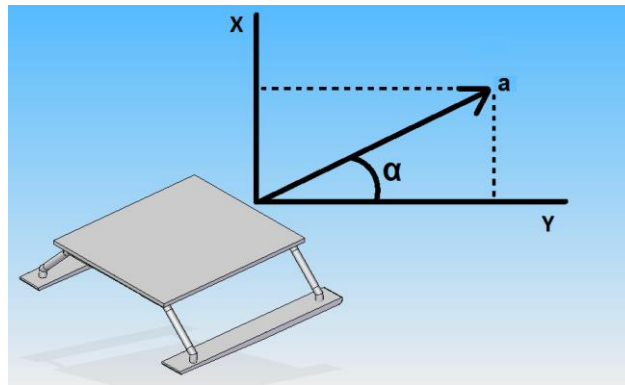
## Modelado dinámico

La mayor parte de los datos son constantes en nuestro problema. Por esta razón podemos unificarlo y expresarlo en función de la velocidad, que es el parámetro que podría llegar a interesar variar, y fijamos los demás valores que dependen de la geometría ya definida.

$$F_T = (1,9 * 10^{-3} + 0,018 + 2 * 1,6 * 10^{-5} + 2 * 0,02 + 0,042) * u^2$$

$$F_T = 0,1 * u^2 \text{ N}$$

Una vez obtenidas las fuerzas no es difícil estimar un modelo dinámico. Teniendo el sistema de coordenadas del plano horizontal podemos descomponer la aceleración en:



31. Descomposición de la velocidad

Uniendo las expresiones de fuerza de fricción, tanto del agua como del aire y el empuje en el plano XY, a la segunda ley de Newton:

$$\sum F = m_T * a$$

$$E - F_f = m_T * a$$

Donde  $E$  es el empuje ofrecido por el motor en Newtons,  $F_f$  es el conjunto de las fuerzas que van en sentido contrario al del avance,  $m_T$  la masa total del sistema y  $a$  su aceleración.

Introduciendo los datos del problema:

$$E - F_{fPlancha} - F_{fFrontalPlancha} - 2 * F_{fPatín-Aire} - 2 * F_{fPatín-Agua} - F_{fEstructura} = m_T * a$$

Que según lo visto anteriormente podemos expresarlo como:

$$E - 0,1 * u^2 = m_T * a$$

Sabiendo que la potencia del motor se puede en función del trabajo:

$$P_M = F_M * r_{giro}$$

Obtenemos la ecuación final:

$$\frac{P_M}{r_{giro}} - 0,1 * u^2 - m_T * a = 0$$

### Servo del timón

Lo siguiente que tenemos que calcular es el par necesario, que tiene que tener el servo para mover el timón. Para ello, ya que el timón estará sumergido en el agua, calcularemos la fuerza que tiene que vencer el timón en el agua por el principio de presión hidrostática.

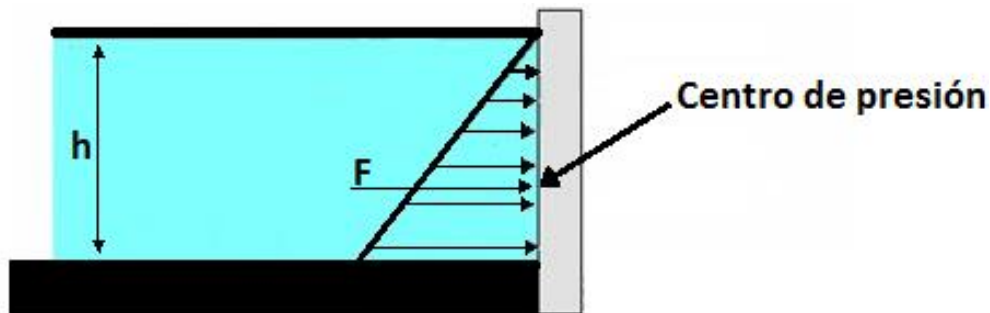
El timón con el que contamos tiene las siguientes características:

$$L_{timón} = 9 \text{ cm}$$

$$A_{LateralTimón} = 64 \text{ cm}^2$$

$$R_{AcciónTimón} = 3,9 \text{ cm}$$

Sabiendo que la distribución de fuerzas de un cuerpo sumergido se produce de la siguiente manera:



32. Distribución de presión

La presión en la interfase tiene que ser la misma, por lo que habrá que igualar la presión atmosférica con la de la superficie del agua. Podemos representar entonces la distribución de presiones como:

$$p + \rho * g * z = p_a + \rho * g * d$$

Que es lo mismo que:

$$p = p_a + \rho * g * (d - z)$$

En este caso, podemos despreciar la presión atmosférica ya que los fabricantes catalogan sus productos por la fuerza a vencer, pero partiendo de la base que están en la atmosfera.

Podemos calcular la fuerza que ejerce esta distribución de presión de la siguiente forma:

$$F_{Timón} = \int_0^d \rho * g * (d - z) * b \, dz$$

Si evaluamos la integral entre  $d$ , que será la longitud del timón y 0 (debido a que tomamos la parte baja del timón como el sistema de referencia  $z=0$ ) nos queda:

$$F_{Timón} = [\rho * g * L * A] - [\rho * g * 0 * A]$$

Que sustituyendo datos nos queda:

$$F_{Timón} = [998 \text{ Kg/m}^3 * 9,81 \text{ m/s}^2 * 0,09 \text{ m} * 6,4 * 10^{-3} \text{ m}^2]$$

$$F_{Timón} = 5,64 \text{ N}$$

Para convertir esta fuerza a un par (T) simplemente hay que multiplicarla por el radio de acción del servo que será el brazo del timón.

$$T = F * R$$

$$T_{servo} = F_{Timón} * R_{AcciónTimón}$$

$$T_{servo} = 5,64 \text{ N} * 0,039 \text{ m}$$

$$T_{servo} = 0,22 \text{ Nm}$$

Como los fabricantes de servos expresan el par en  $\text{Kg/cm}$ , sólo habrá que aplicar un factor de conversión para transformar el resultado.

$$0,22 \text{ Nm} * \frac{1 \text{ Kg}}{9,81 \text{ N}} * \frac{100 \text{ cm}}{1 \text{ m}} = 2,24 \frac{\text{Kg}}{\text{cm}}$$

## Hélice

Ya hemos conseguido el par necesario para comprar un servo acorde al resultado obtenido. Para acabar, lo último que tendremos que calcular es el tamaño de la hélice requerido.

Según el libro *Propeller Handbook* (Gerr, D., 1989) el número ideal de hojas para una hélice sería 1, ya que con una sola hoja no habría otra que le interrumpiese el flujo del fluido. Sin embargo, sería imposible conseguir una estabilidad con una sola hoja, por lo que la siguiente opción a pensar es en 2 hojas. En este caso sí que se consigue la estabilidad necesaria para poder navegar, pero para conseguir un avance efectivo haría falta un diámetro bastante elevado. Por lo que la opción más efectiva es poner 3 hojas, consiguiendo así el equilibrio perfecto entre avance efectivo, equilibrio y diámetro reducido.

Una vez decidido el número de hojas, ya solo falta calcular el diámetro mínimo necesario para nuestro caso, para ello usaremos la fórmula que viene a continuación obtenida también del libro mencionado en el párrafo anterior:

$$F_T = \frac{\pi}{4} * D^2 * \left( V + \frac{\Delta V}{2} \right) * \rho * \Delta V$$

Donde:

$$F_T = \text{Fuerza total de arrastre (N)}$$

$$D = \text{Diámetro de la hélice (m)}$$

$$V = \text{Velocidad del fluido (m/s)}$$

$\Delta V$  = Velocidad incrementada por la hélice ( $m/s$ )

$\rho$  = Densidad del medio ( $Kg/m^3$ )

Despejando el diámetro de la hélice que es lo que queremos calcular, nos queda:

$$D = \sqrt{\frac{4 * F_T}{\pi * \left(V + \frac{\Delta V}{2}\right) * \rho * \Delta V}}$$

Consideraremos el flujo por el que circulamos estacionario, ya que simplemente queremos calcular un diámetro mínimo, al que luego le añadiremos un margen para estar previstos para cuando el flujo de agua esté en movimiento.

$$D = \sqrt{\frac{4 * 2,55 \text{ N}}{\pi * \left(\frac{5 \text{ m/s}}{2}\right) * 998 \text{ Kg/m}^3 * 5 \text{ m/s}}} = 0,016 \text{ m}$$

Hemos obtenido que:

$$D = 1,6 \text{ cm}$$

Por lo que tomaremos este dato como diámetro mínimo necesario para el desplazamiento.

## 8. Hardware

Una vez finalizada la parte estructural, es necesario proceder con la parte electrónica.

### 8.1 Sistema de control

#### Sensores de obstáculos

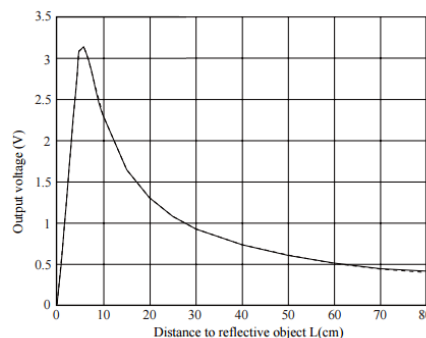
Lo primero que hay que pensar es en el sistema de navegación es en la evasión de obstáculos, para ellos necesitaremos un dispositivo que se encargue de reconocer los objetos que puedan interferir en la trayectoria de la base, al principio fueron varias las posibilidades barajadas.

La primera fue un sensor de infrarrojos como el de la figura de abajo.



33. Sharp GP2Y0A21YK0F (Watterott, 2009)

Una vez investigado el funcionamiento de este tipo de sensores fue rápidamente descartado. Ya que funcionan con un haz de luz que es enviado por un led y recibido por el otro. Esto se produce debido a la reflexión del haz en una superficie sólida por lo que no era recomendable para una situación en la que el medio es mayoritariamente acuático. El agua absorbe cierta parte de esta luz y, además, varía su ángulo por lo que podría provocar medidas erróneas o directamente que el haz no llegase a su receptor produciendo que no se produjese ninguna medida de distancia.



34. Alcance del Sharp GP2Y0A21YK0F (Datasheet, 2006)

Como vemos, a pesar de depender de la superficie de reflexión, aun encontrándose en un medio favorable la distancia de alcance de estos sensores, vemos en el gráfico anterior que es bastante reducida, por lo que no es recomendable en nuestro caso.

Las características que diferencian a unos sensores ultrasónicos de otros son el alcance y el ángulo efectivo de medición. El alcance no suponía un problema ya que todos conseguían el mínimo que podríamos considerar necesario para nuestro caso. El problema sí que era el ángulo efectivo de medición, ya que necesitamos que la base esté libre de obstáculos en sus 360°. Existen sensores de gran angular, pero estos llegan a alcanzar en el mejor de los casos unos 80°, por lo que sería necesario colocar un gran número de sensores para cubrir todo el exterior. Descarté esta opción debido al elevado precio de estos sensores en comparación con los de medición estándar, por lo que finalmente encontré una solución alternativa: colocar el número mínimo de sensores pero con un sistema de barrido para que con un ángulo menor podría cubrir la misma superficie, por ello elegí uno de los sensores más económicos para reducir costes, los cuales han sido:



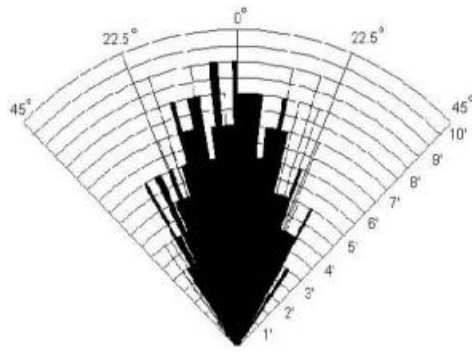
35. HC SR04

Características	
Alimentación	+5v DC
Frecuencia de trabajo	40 KHz
Consumo (suspendido)	< 2mA
Consumo (trabajando)	15mA
Ángulo efectivo	< 15°
Distancia	2cm a 400cm *
Resolución	0.3 cm

\*A partir de 250cm la resolución no es buena

36. Tabla de características HC SR04 (Zonamaker, 2016)





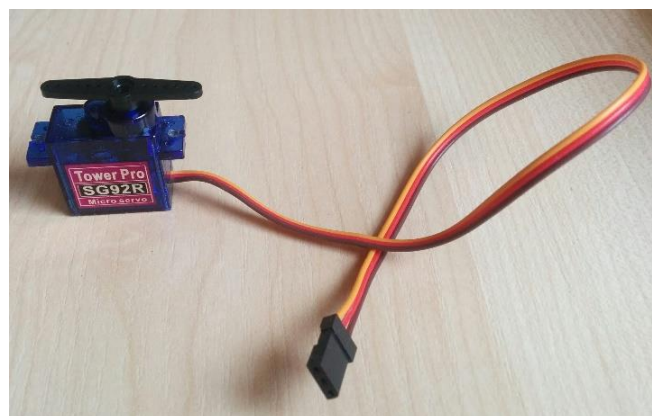
37. Test de alcance (Datasheet, 2010)

Como vemos en la tabla de características de dicho sensor el ángulo efectivo es muy pequeño, no más de 30 grados para asegurar que no hay errores de precisión. Vemos en el gráfico de alcance que para zonas más lejanas la amplitud de medición se va reduciendo considerablemente por lo que perdemos fiabilidad en las medidas. Por ello, la solución que se ha adoptado es montar este sensor en un servo que se encarga de variar su orientación para hacer un barrido de 90 grados. Colocando uno en el centro de cada lado del rectángulo que es la base, nos aseguramos de cubrir todo el perímetro. Teniendo 4 sensores que cubren 120 grados cada uno (sumando los 30 ya del sensor) nos va a producir que cierta área de medición se solape con su sensor más próximo. Sin embargo, podemos ver esto como algo positivo, ya que nos asegura que el perímetro estará cubierto, al menos en sus esquinas, durante el mayor tiempo posible.

### Servos

El servo usado para hacer el barrido ultrasónico ha sido el de la figura de abajo. No es necesario que tenga un par elevado, ya que el esfuerzo a realizar es mínimo, y como su única función es mover en un ángulo de menos de 180° el sensor de ultrasonidos, cuyo peso es despreciable (10 gramos) se ha adquirido el más pequeño disponible en el proveedor para esta función.

Estos servos se encargarán de hacer un barrido de 45 a 135 grados (no un barrido de 180 grados para evitar interferencias con la estructura) para tomar información del entorno completo del sistema.



38. Servo SG92R

CARACTERÍSTICAS	
Par	2,5 Kg/cm
Velocidad	0,08s/60º
Rango de rotación	180º
Peso	9 g
Dimensiones	23 x 12 x 27 mm
Voltaje	4,8 - 6 V

Además, será necesario un servo más encargado de mover el timón. En este caso se trata de un servo más potente y además capaz de resistir el agua, ya que al tener que encontrarse cerca de la superficie del agua, deberá estar preparado para posibles salpicaduras. Una vez investigado los modelos disponibles, este fue el elegido:



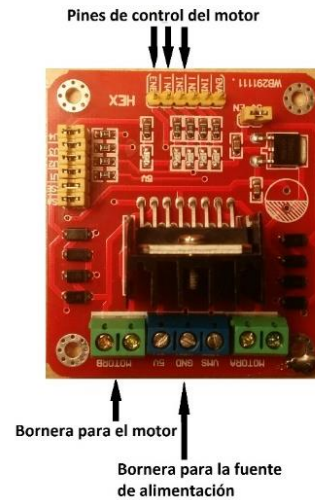
39. Servo waterproof

CARACTERÍSTICAS	
Par	14 Kg/cm
Velocidad	0,17s/60º
Rango de rotación	300º
Peso	60 g
Dimensiones	40 x 20 x 41 mm
Voltaje	6 – 7,2 V

En este caso el servo elegido cumple con un par de torsión mucho mayor al requerido. En el apartado 7 obtuvimos que, el par necesario para el timón es de 2,24 *Kg/cm*, pero al necesitar un servo con características especiales (a prueba de agua), esto es lo más cercano a nivel comercial en los proveedores habituales.

### Controlador del motor

Para controlar el motor se usará un módulo de Arduino conocido como L298N. Este *shield* es capaz de controlar el funcionamiento del motor, así como su sentido y velocidad. Tiene dos salidas para poder controlar dos motores diferentes, pero en este caso solo es necesario un motor.



40. L298N

En un principio el propuesto fue el módulo L293D, es muy similar a este salvo por una excepción: el límite de corriente era de 1,2A para cada uno de los motores, pero necesitamos en nuestro caso motores de más potencia debido a las dimensiones de la construcción. Por lo que finalmente hubo que usar el L298D, el cual permite un paso de corriente de hasta 2A, más adecuado para el motor a usar en este proyecto.

Para las conexiones de este módulo se necesita:

- Conexión a Arduino: 3 cables. EN3 y EN4 para elegir el sentido de giro del motor (Estos ya que el motor va conectado a la bornera B, si hubiese sido la del A las conexiones hubieran sido en EN1 y EN2). Cuando uno está activado hay que desactivar el otro para elegir un sentido de giro, y al revés para el contrario. El otro pin a conectar es ENB, el cual activa o desactiva el motor B.
- Alimentación. En este caso hay dos posibles opciones: en el caso de usar una fuente de 5V, será necesaria una entrada de corriente en VMS para el motor y conectar el puente selector de 5V para que sea esta misma fuente la que alimente también la lógica del circuito. En caso de desconectar este puente, se usará una entrada de corriente en VMS para el motor de 6-12V y otra entrada de 5V en el pin con el mismo nombre para el circuito. En este caso, usaremos esta última opción para poder usar una fuente de más potencia en el motor.

## Motor

El módulo anterior se encargará de controlar el motor elegido:



41. Motor

CARACTERÍSTICAS	
Voltaje	7,2 V
Consumo a plena carga	6 A
Potencia	43,2 W
Velocidad angular	13600 rpm
Peso	73 gr

En este caso el motor elegido tiene algo más de potencia que requerían los cálculos. Por lo que tendremos margen para pérdidas de potencia.

## Arduino

Todo esto será controlado desde un microcontrolador programable, en este caso se ha optado por un Arduino. Debido al número de entradas digitales requeridas, se ha tenido que usar un Arduino Mega, ya que el Arduino Uno se limita a 14 entradas digitales, lo que sería insuficiente para este proyecto.

También existía la posibilidad de incorporar un multiplexor de pines al Arduino Uno, uno como el de la imagen siguiente:



42. Multiplexor de pines Arduino (Sparkfun, 2016)

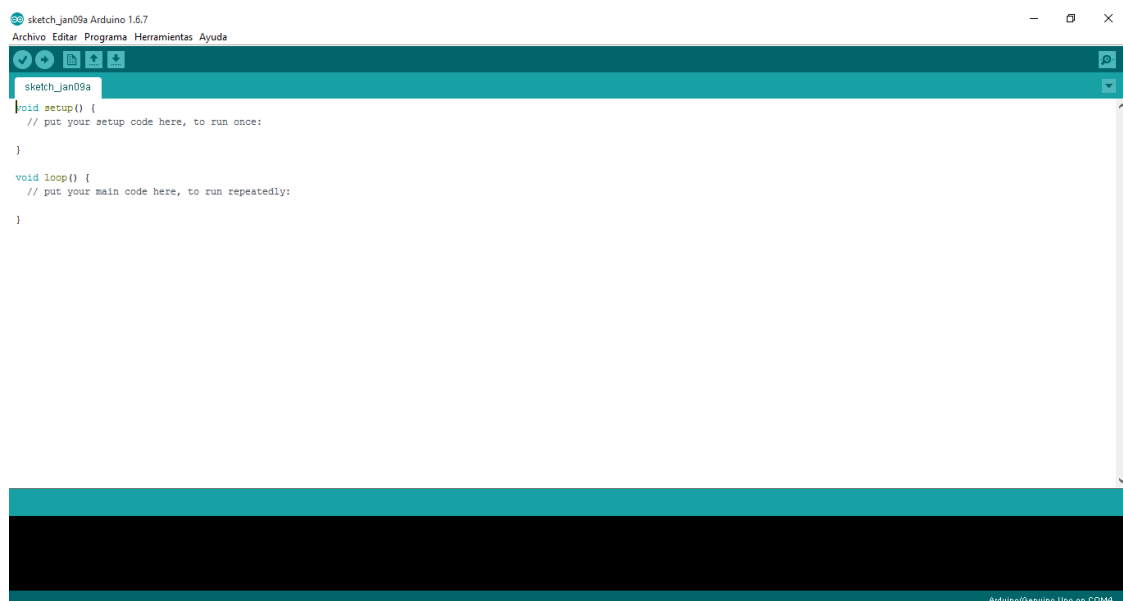
Esta opción parecía interesante, pero fue descartada después de un pequeño análisis. Para empezar, hace falta construir un conector de pines para unir los básicos de Arduino a estos. Para ello es necesario adquirir el conector y realizar algo de mano de obra, además del ya elevado precio del propio multiplexor. También, complicaría bastante la infraestructura del microprocesador, por lo que finalmente la opción elegida ha sido:



43. Arduino Mega 2560

De esta forma contaremos con 54 entradas/salidas digitales, con lo que será más que suficiente para este propósito.

Para la programación de este Arduino se ha usado el software oficial de dicha marca. En él, aparecen dos partes diferenciadas: el módulo “setup” donde se escribirá todo aquello que se deba ejecutar una sola vez al principio del programa, como pueden ser la definición de los modos de uso de los pines; y el módulo “loop” que será donde se escriba el programa en sí, lo que haya dentro de esto se ejecutará continuamente. Antes de estas partes, al comienzo del programa, se declararán las variables y las constantes.

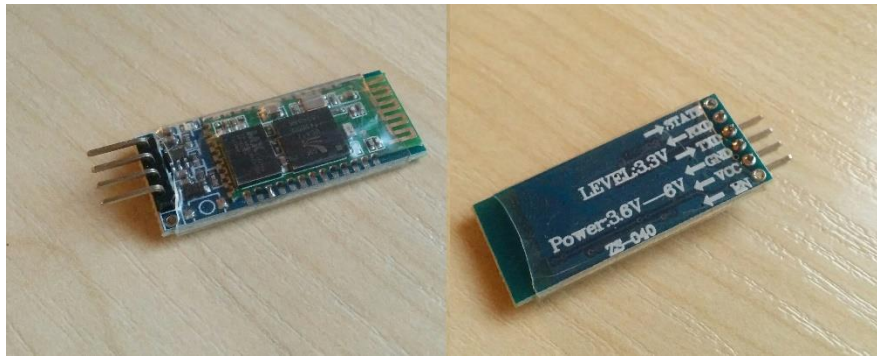


44. Programa Arduino



## Bluetooth

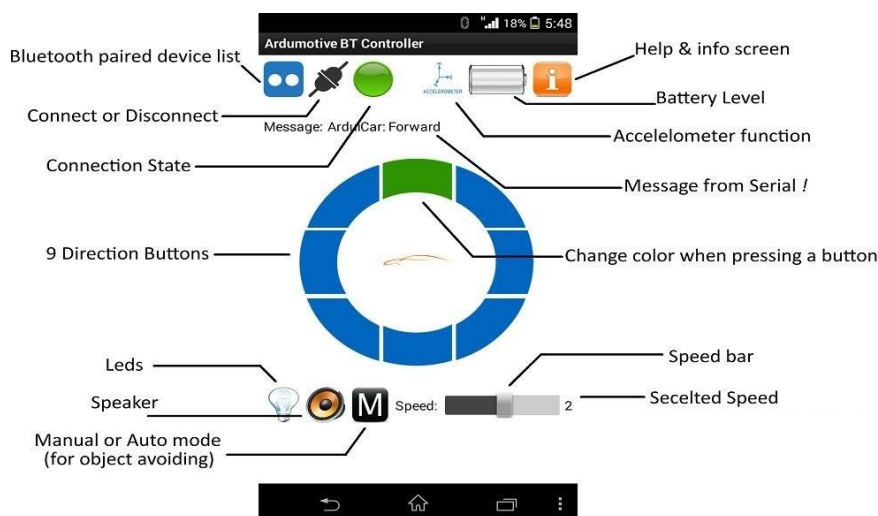
También es necesario dotar a la base de un sistema de control a distancia. Esto es debido a que una vez en el agua, aparte de tener el sistema de control automático en el que la balsa navegará sola por el cauce del agua con la única directriz de evitar los posibles obstáculos que puedan surgir, es necesaria la implantación del módulo HC 06, se trata de un módulo de comunicación Bluetooth para poder controlar el sistema mediante control remoto a un alcance de unos 25-30 metros.



45. Módulo Bluetooth HC 06

Como dispositivo de control usaremos un Smartphone, a través de una aplicación diseñada para el control mediante Bluetooth de Arduino. Dicha aplicación puede ser encontrada por *Ardumotive BT Controller*. Inicialmente está diseñada para controlar vehículos terrestres, pero con unas ligeras modificaciones en el código se ha adaptado para poder ser usado en este caso.

Para que la comunicación sea posible es necesario introducir la contraseña por defecto "1234" en el dispositivo móvil. Una vez hecho, el led rojo del módulo bluetooth comenzará a parpadear hasta que ambos estén conectados, en ese momento el led pasará de parpadear a quedarse fijo indicando que la conexión se ha realizado correctamente.



46. Aplicación (Ardumotive, 2013)

Aquí vemos una captura de la aplicación en la que podemos ver las principales funciones disponibles.

Hay algunas funciones que no se han habilitado: como son el indicador de batería y las funciones de Luz y Bocina, ya que en este caso no procedía. Por lo que se ha usado estos dos iconos para hacer un sistema para poder alternar entre modo manual de navegación y modo automático.

Otra opción de comunicación hubiera sido por radiofrecuencia, a través de un emisor y receptor como los de la figura de abajo se podría haber logrado también la comunicación entre el móvil y la base. Sin embargo, no se usó esta opción ya que la comunicación mediante este tipo de tecnología es más pobre y débil, por lo que al no ser mucho más caro el módulo bluetooth, esta última opción fue elegida.



47. Módulos RF (Dx, 2016)

## Alimentación

En cuanto a la parte de alimentación, Arduino será alimentado con la batería directamente, ya que cuenta internamente con un transformador que admite hasta 12V. De esta forma, podemos unificar toda la alimentación en un único elemento.



48. Batería LRP



Las características elegidas para la batería han ido acorde con el motor comentado anteriormente. Se trata de una batería NiMh de 4 amperios a 7,2V. Para reponer la energía de esta batería ha hecho falta un cargador adecuado para este tipo de fuentes de alimentación, lo podemos observar en la siguiente imagen:



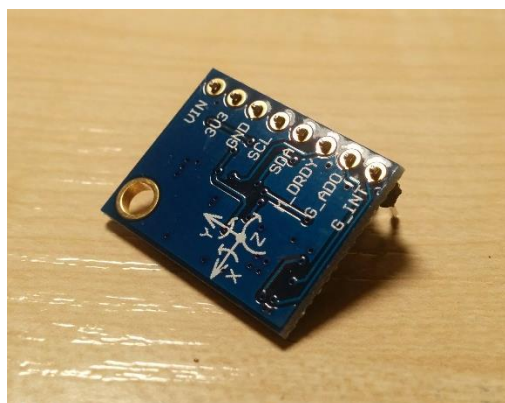
49. Cargador batería LRP

Una vez conectado todo esto de forma correcta, incluyendo cableado y código necesario (Incluido en el Anexo), obtenemos el módulo de control.

## 8.2 Sistema de orientación y aceleración

### IMU

Para el cálculo de la velocidad se usará una IMU (*Inertial Measurement Unit*), la cual está compuesta por un acelerómetro (ADXL345), un giroscopio (ITG3205) y una brújula (HMC5883L). En nuestro caso nos centraremos en el uso del acelerómetro para medir la velocidad de avance de la balsa.



50. IMU GY-85

El acelerómetro mide la aceleración lineal en cada eje. En el eje Z (eje vertical) siempre contaremos con la aceleración de la gravedad (9.8 m/s), no obstante se puede añadir un offset fácilmente para obtener una medida de la aceleración en este eje relativa.

En este caso, se centrará el uso de este dispositivo en el eje Y, el cual se colocará en el sentido de avance de la base de aterrizaje. La medición de la velocidad no es inmediata, ya que con esto obtenemos la aceleración, por lo que habrá que hacer unos pasos previos a la obtención del parámetro deseado.

Tomaremos el tipo de movimiento descrito por el prototipo como MRUA (Movimiento Rectilíneo Uniformemente Acelerado) y usaremos sus fórmulas, en concreto esta.

$$x = x_0 + V_0 * t + \frac{1}{2} * a * t^2$$

Donde:

$x$  = Posición en el sistema de referencia fijo (m)

$x_0$  = Posición en el sistema de referencia móvil (m)

$V_0$  = Velocidad en el instante  $t$  (m/s)

$t$  = Tiempo (s)

$a$  = Aceleración ( $m/s^2$ )

Como se toma el sistema de referencia fijo en el móvil tenemos que  $x = x_0$ . Despejando la velocidad de la ecuación resultante:

$$V_0 = \frac{a * t}{2}$$

El signo negativo se desprecia ya que este únicamente indicaría el sentido positivo del eje, que podemos considerar el más conveniente en nuestro caso.

Se implementará esta fórmula en el código de Arduino para conseguir la expresión instantánea de la velocidad en cada momento.

También ha sido necesario ajustar el *Offset* de este componente, ya que de fábrica viene desajustado y da medidas incluso en estático. Este problema se soluciona con sumándole o restándole la cantidad que se desvía de cero, consiguiendo así una lectura de cero en parado.

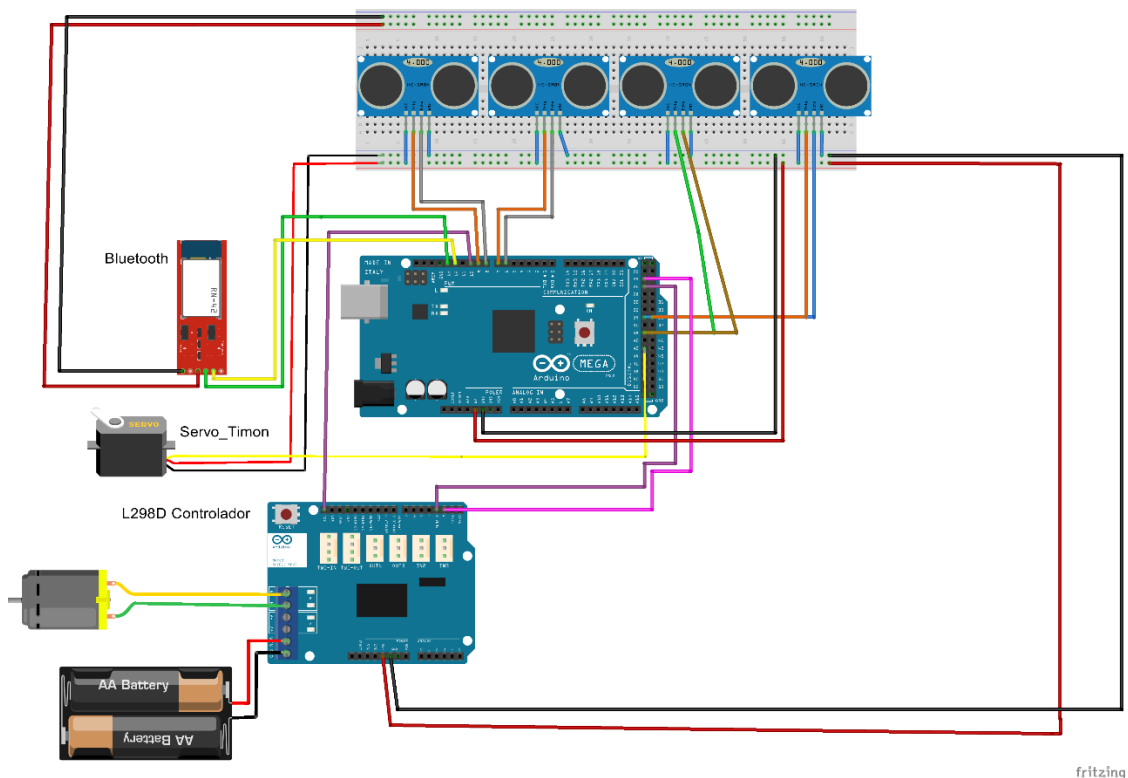
Además, también se implementará el código necesario para la lectura de los grados de la brújula, esto proporciona la posibilidad de conocer en todo momento la orientación de la balsa. Esto es realmente útil al realizar la medición de parámetros en el archivo de datos, permitiéndonos conocer el giro ocasionado como consecuencia de una acción.

### 8.3 Esquema eléctrico

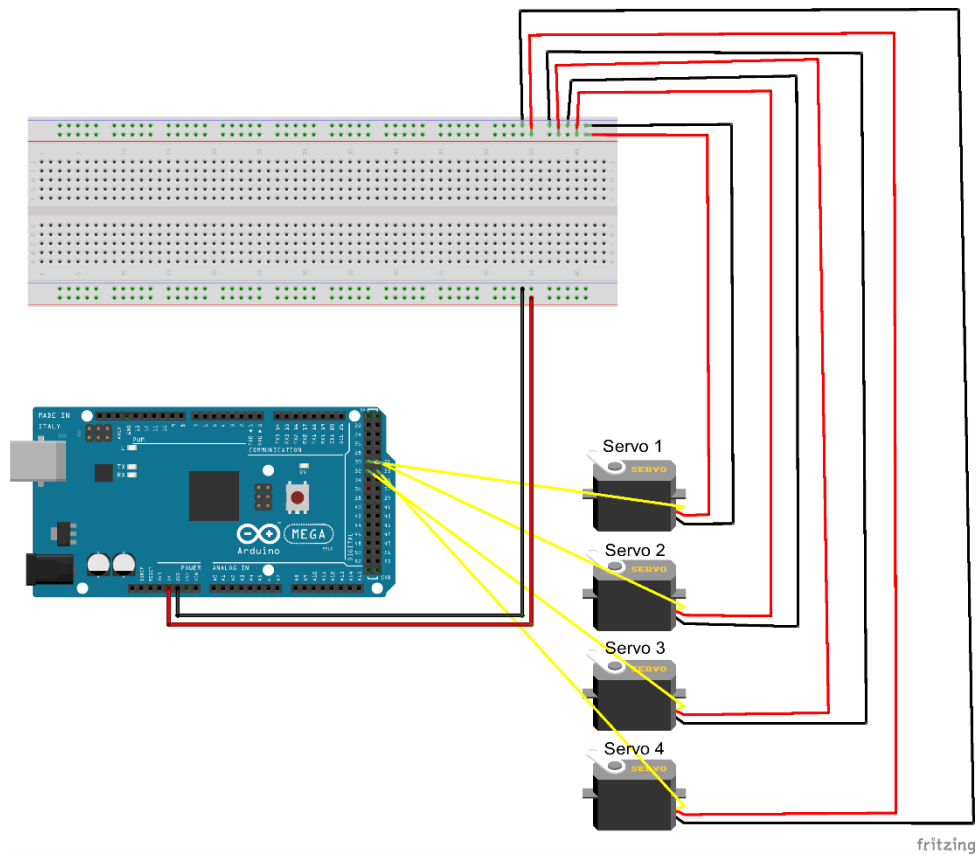
Es necesario realizar correctamente la conexión de todos los elementos anteriores. Hay que tener en cuenta varios criterios para ello:

- Conectar entre A0 y A15 todos los componentes que requieran el uso de señales analógicas para su control y lectura.
- Usar los pines comprendidos entre el 2 y el 14, además del 44 y el 46 para todos aquellos dispositivos que necesiten modulación PWM para su funcionamiento.
- Del 14 al 19 para comunicación y el 20 y 21 para comunicación SCL y SDA.
- En los demás pines irán conectados los que requieran de un tratamiento digital.
- Por supuesto, es necesaria la correcta asignación de los polos positivos con la alimentación positiva y los polos negativos con GND.

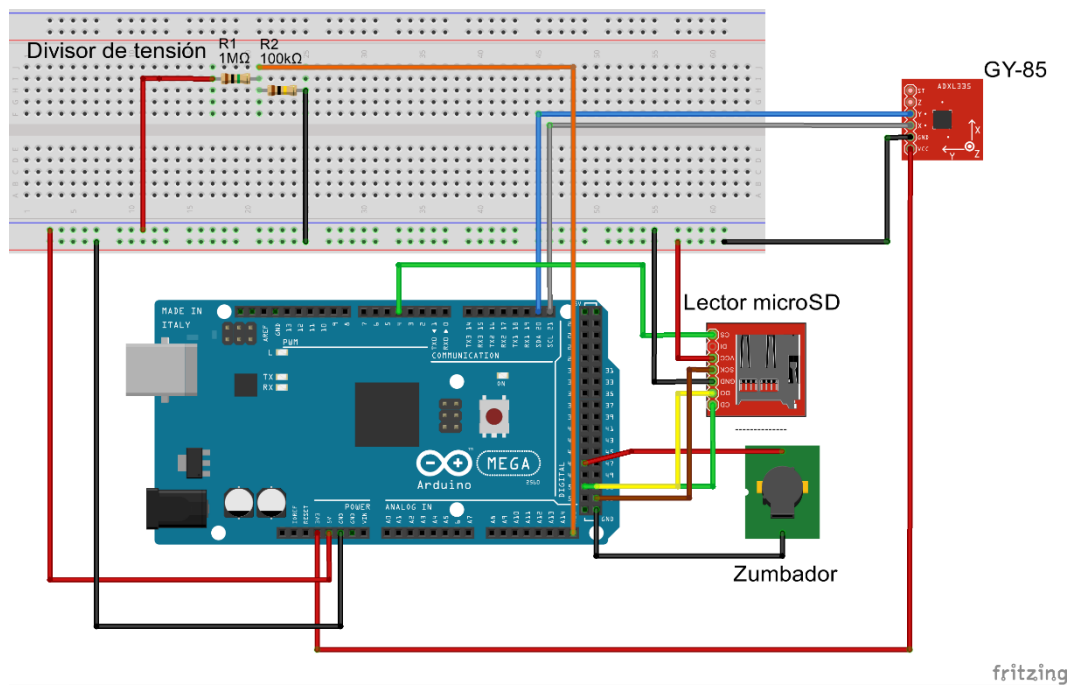
Cumpliendo estos requisitos vemos a continuación la interconexión de los distintos sistemas que componen este proyecto. Debido a la complejidad del circuito, se expondrá dividido en 3 secciones.



51. Sistema de control



## 52. Sistema de barrido



### 53. Otros componentes

En la página anterior tenemos una visión global de toda la electrónica que porta el proyecto. Podemos describir la relación de conexiones como:

- Sensor ultrasonidos Izquierdo: VCC -> 5 V  
GND -> Tierra  
Trig -> Pin 35  
Echo -> Pin 34
- Sensor ultrasonidos Delantero: VCC -> 5 V  
GND -> Tierra  
Trig -> Pin 37  
Echo -> Pin 36
- Sensor ultrasonidos Trasero: VCC -> 5 V  
GND -> Tierra  
Trig -> Pin 7  
Echo -> Pin 6
- Sensor ultrasonidos Derecho: VCC -> 5 V  
GND -> Tierra  
Trig -> Pin 9  
Echo -> Pin 8
- Servo del Timón: VCC -> 5 V  
GND -> Tierra  
Control -> Pin 42
- Bluetooth: VCC -> 5 V  
GND -> Tierra  
RX -> Pin 13  
TX -> Pin 12
- IMU (GY85): VCC -> 3.3 V  
GND -> Tierra  
SDA -> Pin 20  
SCL -> Pin 21
- L298N: VCC -> 5 V  
GND -> Tierra  
VMS -> Fuente de alimentación externa (Negativo a GND también)  
IN3 -> Pin 26  
IN4 -> Pin 24  
INB -> Pin 10 (PWM)  
Bornera B -> Motor

- Servo barrido Izquierda: VCC -> 5 V  
GND -> Tierra  
Control -> Pin 30
- Servo barrido Delantero: VCC -> 5 V  
GND -> Tierra  
Control -> Pin 31
- Servo barrido Trasero: VCC -> 5 V  
GND -> Tierra  
Control -> Pin 32
- Servo barrido Derecho: VCC -> 5 V  
GND -> Tierra  
Control -> Pin 33
- Zumbador: VCC -> 5 V  
GND -> Tierra
- Lector SD: VCC -> 5 V  
GND -> Tierra  
MISO -> Pin 50  
MOSI -> Pin 51  
SCK -> Pin 52  
CS -> Pin 53

## 8.4 Componentes de navegación y transmisión

### Transmisión

Para transmitir el movimiento de la estructura electrónica al agua son necesarios distintos componentes. Primero, será necesario transmitir el movimiento del motor a través de un eje, en este caso necesitamos una varilla de acero de 4mm y un tubo de cobre de 6mm con 4 mm de interior.



54. Componentes del eje de transmisión

Introduciendo la varilla dentro del tubo con una cobertura de grasa y con los prisioneros fijados en los extremos conseguiremos la construcción del eje. Uniremos este eje al motor mediante un conjunto cardán:



55. Cardán

### Reductora

Ha sido necesaria la incorporación de un elemento adicional al sistema de transmisión, se trata de una reductora 3:1.



56. Reductora 3:1

La función de este componente es la de aumentar el par de salida del motor a costa de reducir la velocidad. Su incorporación nos permite aumentar la fuerza del motor, lo cual nos es útil debido al alto peso de la balsa, bajando la velocidad de rotación del motor, lo cual no es ningún inconveniente ya que no necesitamos un desplazamiento rápido para nuestro objetivo.

Con un breve cálculo podremos ver cuál es el aumento de par provocado por este componente. Primero, calcularemos cuál es el par actual:

$$Potencia = Par \times Velocidad\ angular$$

$$Par = \frac{Potencia}{Velocidad\ angular}$$

$$Par = \frac{43,2\ W}{13600rpm * \frac{2 * \pi}{60}} = 0,03\ Nm$$



Como la reductora posee una relación de 3 a 1, estos serán los nuevos parámetros del motor:

$$Par_{reductora} = Par * 3 = 0,03 Nm * 3 = 0,09 Nm$$

$$Velocidad\ angular_{reductora} = \frac{Velocidad\ angular}{3} = \frac{13600\ rpm}{3} = 4533,33\ rpm$$

### Hélice

En el extremo opuesto al motor se encontrará la hélice, en este caso hemos usado una de 3 álabes de 40cm de diámetro.

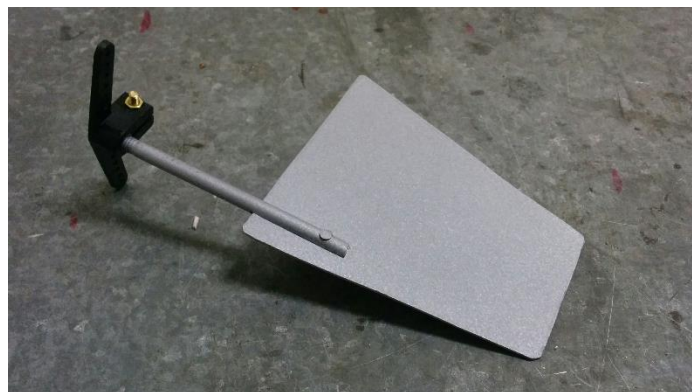


57. Hélice PVC 40mm

En los cálculos obtuvimos un diámetro necesario para la hélice de 1,6cm, pero sin tener en cuenta perturbaciones en el medio, ni velocidad a contra corriente ni fluctuaciones en el nivel del agua, por estos motivos y por las grandes dimensiones del proyecto, finalmente el diámetro usado ha sido de 40mm.

### Timón

Hasta aquí se ha expuesto el conjunto de piezas encargado de transmitir el movimiento rotativo del motor hasta el agua. La otra parte importante en este apartado la encargada de modificar la dirección del movimiento de la base, para ello usaremos, unido al servo destinado para ello, el timón, el cual podemos ver en la siguiente imagen:



58. Timón

En este caso no disponían en el proveedor del timón adecuado que el proyecto requería, ya que necesitaba un área bastante más grande de lo que tenían en stock, por lo que ha sido necesario realizar uno casero.

## 8.5 Otros elementos necesarios

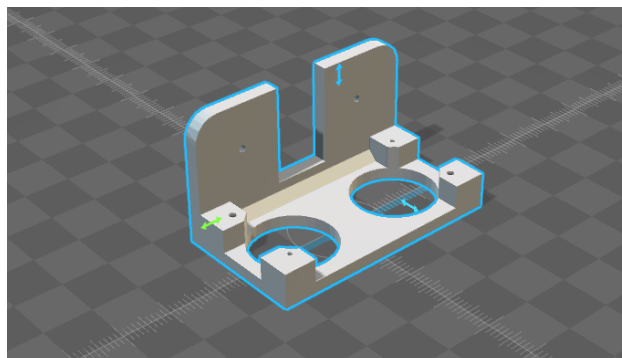
### Soporte servo de barrido

Los sensores ultrasonidos van montados en un sistema de barrido, el cual es usado para ampliar el campo de actuación de estos. Se compone de un soporte metálico de fabricación casera encargado de unir el servo a la superficie inferior de la balsa. El servo va fijado al soporte por dos tornillos y este unido con cinta de doble cara a la balsa.



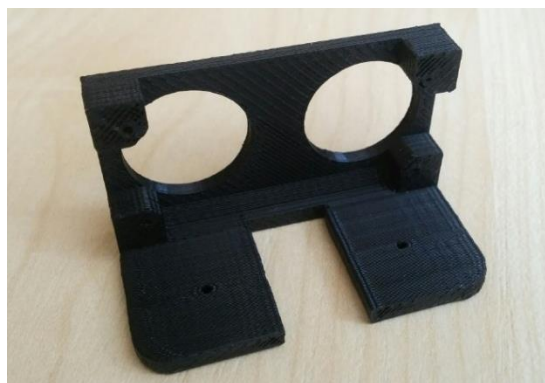
59. Soporte del sensor ultrasonidos

Por otra parte, es necesario otro soporte que una el sensor ultrasonidos con la parte móvil del servo. En este caso, el diseño ha sido obtenido de un servidor online (Thingiverse) en la que se compartían diseños adaptados y dimensionados para un objetivo en concreto preparado para ser usado en una impresora 3D, el elegido para este propósito se puede ver a continuación.



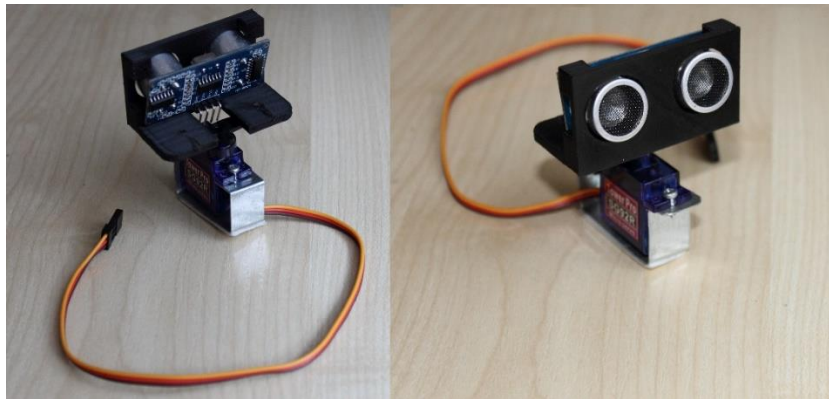
60. Diseño 3D soporte ultrasonidos (Thingiverse, 2013)

Después de procesar este diseño con una impresora 3D, este fue el resultado obtenido:



61. Impresión 3D

Siendo este el resultado total obtenido del conjunto.



62. Conjunto soporte servo-HC04

### Interruptor

Para controlar el encendido/apagado del sistema se usará un interruptor como el de la imagen de abajo. Estará ubicado en la parte exterior de la caja de electrónica.



63. Interruptor de corriente

### Zumbador

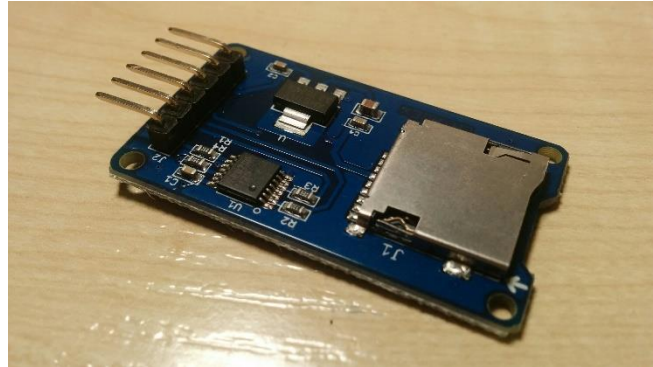
Otro elemento que ha sido necesario incorporar ha sido un zumbador. Para asegurar una vida longeva a la batería es conveniente monitorizar su caída de potencial. En nuestro caso tenemos una batería de 7.2 V, tenemos que asegurar que este voltaje no baje de 6V, ya que en caso de que esto sucediese podría llegar a provocar su ruptura. Para ello se ha incorporado un zumbador el cuál se activará cuando la batería baje de los 6V. Cuando esto pase, hay que desconectar el sistema y conectarlo a su cargador hasta que esté completamente cargada.



64. Zumbador

### **Lector de tarjetas microSD**

Para corroborar el correcto funcionamiento del prototipo, usaremos un lector de tarjetas microSD para monitorizar todas las medidas tomadas y cálculos realizados durante la ejecución del módulo de navegación automática para su estudio posterior.



*65. Lector de tarjetas microSD*

En él se registrará velocidad, orientación de la balsa, la distancia medida por los sensores, el ángulo del timón en cada momento y la acción desencadenada.

## 9. Construcción del prototipo

Lo primero para comenzar el proceso de montaje del sistema será la incorporación del motor en la caja de componentes electrónicos, usaremos para ello la reductora como soporte. Una vez introducido el motor en la reductora utilizaremos esta como anclaje del conjunto a la superficie interna de la caja.



*66. Montaje del motor y la reductora*

Con la ayuda de unas cuñas de madera dotaremos al eje de salida la inclinación necesaria para favorecer la transmisión del movimiento. Este paso es recomendable ya que sin él, el cardán trabaja con un ángulo demasiado elevado, y además de forzar su estructura, la fuerza necesaria para mover el eje aumenta de forma considerable debido al ángulo de acción.

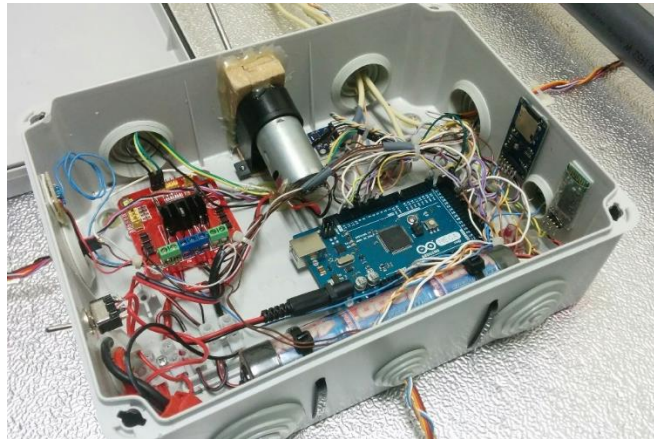
El eje del motor asomará por la parte exterior de la caja, donde se colocará el cardán. Para ello quitaremos uno de los tapones de goma de la parte estrecha de la caja y usaremos este orificio como salida de la parte mecánica al exterior. Se puede observar el resultado en la siguiente imagen:



*67. Unión cardán-eje*

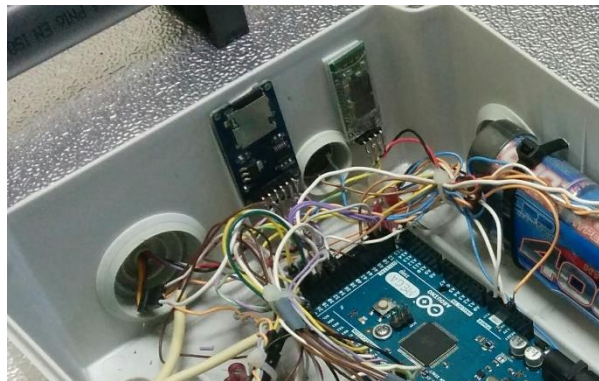


Después fijaremos el hardware necesario en la caja y realizaremos las conexiones pertinentes:



*68. Caja de conexiones*

Algunos componentes han sido unidos a las paredes de la caja mediante cinta de doble cara.



*69. Componentes adicionales*

Para las conexiones de los polos positivos y negativos se han usado clemas para unificar los cables y evitar los empalmes, de esta manera se facilita la posibilidad de modificaciones futuras.



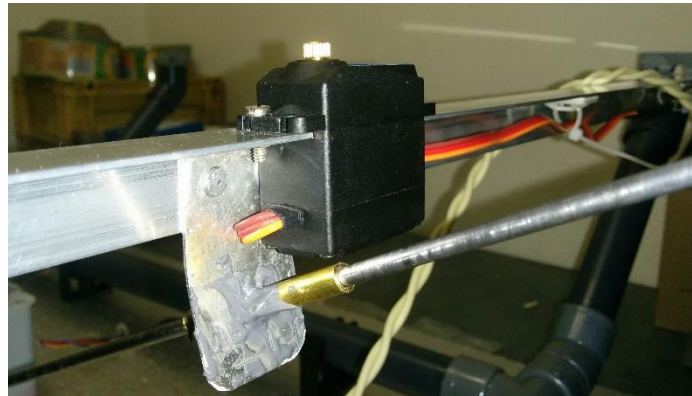
*70. Clemas de conexión*

Después de encontrar el punto medio de cada lateral de la base, se colocarán los soportes de los servos de barrido. Su adhesión se realiza con cinta de doble cara.



*71. Instalación del soporte del servo de barrido*

Para unir el sensor ultrasonidos al soporte se ha usado pegamento termofusible. Después de esto, resta por instalar el servo del timón. Se ha realizado un cajeado en la pletina de aluminio y con dos tornillos se consigue su perfecta unión.



*72. Instalación del servo del timón*

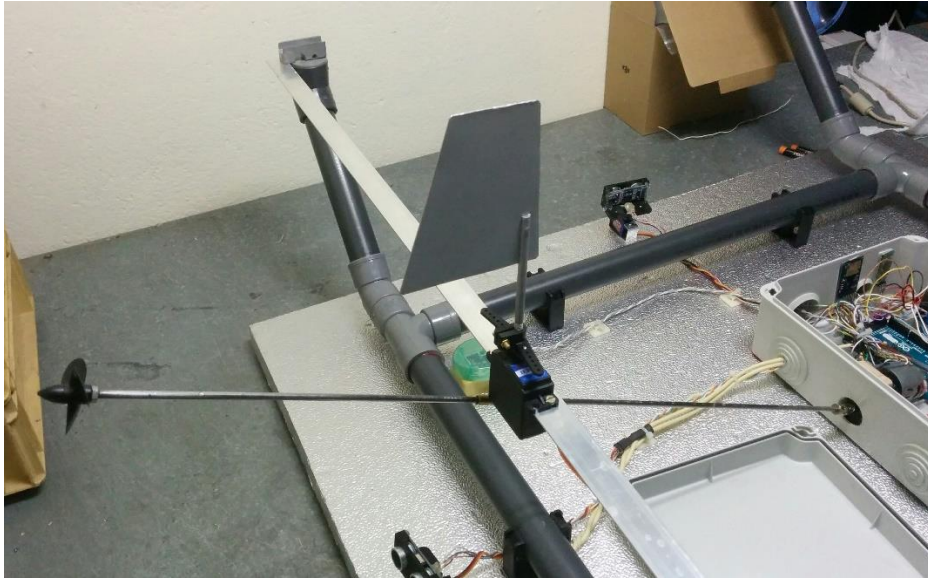
En la foto también aparece el casquillo del eje. Este casquillo se encarga de hacer de punto de apoyo al eje de giro. En un principio el casquillo ocupaba toda la longitud del eje, no obstante, esto incrementaba el rozamiento en el conjunto casquillo-eje. De la forma que aparece en la foto, la fuerza necesaria es mucho menor consiguiendo un aumento del rendimiento. Viéndose así desde la parte final del eje:



*73. Vista del eje*



El timón va encajado en el servo de dirección:



*74. Instalación del timón*

Otra adaptación que ha habido que realizar es la ubicación del zumbador, para escuchar la bocina de la parte exterior tiene que ubicarse en la parte de fuera de la caja, siendo esta la adaptación elegida:



*75. Adaptación del zumbador*

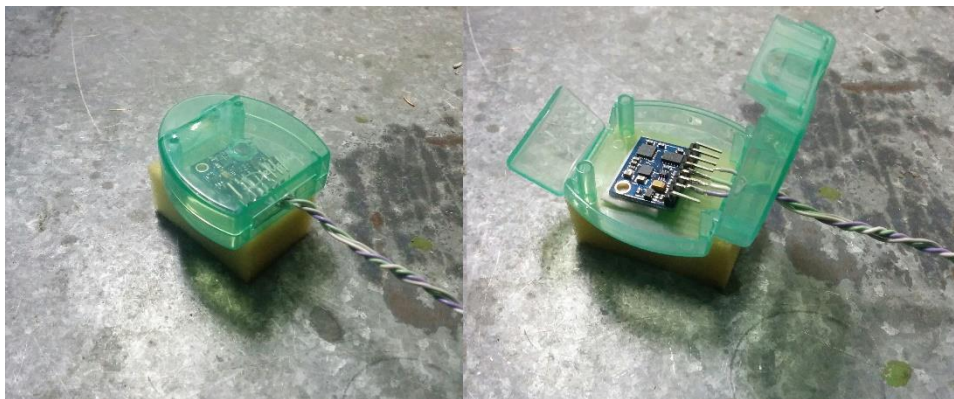
Encastrado en el tapón de goma nos aseguramos que no se pierde la hermeticidad.

También ha habido que colocar un interruptor de encendido.



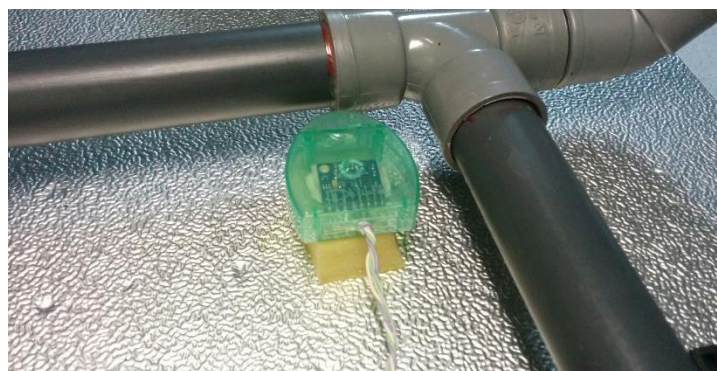
*76. Interruptor de puesta en marcha*

Debido a las características del acelerómetro ha sido necesaria realizar una instalación específica para él. Es un componente muy sensible a perturbaciones del medio, como por ejemplo a las vibraciones y a los campos magnéticos. Por ello, se ha instalado en una cápsula de plástico aislada con una esponja para la absorción de las oscilaciones.



*77. Encapsulado del acelerómetro*

El motor es un gran generador tanto de vibraciones como de campos magnéticos, como estas contaminan toda la caja, se ha realizado la instalación del acelerómetro fuera de su perímetro.



*78. Ubicación del acelerómetro*

Finalmente, esta es la forma de la estructura después de la instalación de la parte electrónica:



*79. Instalación electrónica*



*80. Resultado de la instalación*

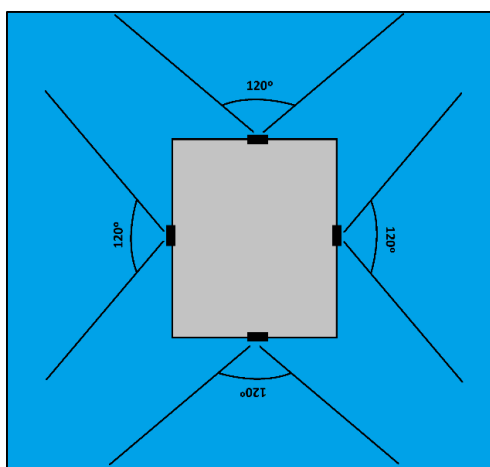


## 10. Módulo de evasión de obstáculos

En un principio se optó por una lógica de programación que después fue versionada para mejorar su funcionamiento. Comienza esta parte con la descripción de dicho primer método.

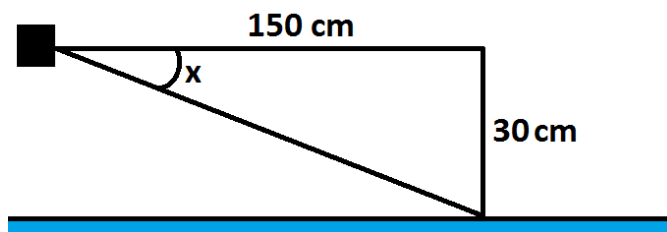
### Primer método de programación (No reactivo)

Para la percepción de obstáculos en el agua se han colocado 4 sensores ubicados de forma estratégica. Colocando uno en el centro de cada lado de la base y con un sistema de barrido para cada sensor, se consigue en cubrir toda el área perimetral. 30 grados proporcionados por el sensor, y otros 90 que se suman gracias al barrido del servo.



81. Campo de acción horizontal de los sensores

En esta imagen hemos visto la cobertura horizontal, ahora trataremos la vertical. Los sensores solo disponen de un campo efectivo de 30°, por lo que para cubrir una superficie del agua cercana deberemos dotar a los sensores de cierta inclinación. Podemos calcular dicha inclinación sabiendo que, la altura de la superficie del agua al sensor es de 30cm y además, debemos asumir la distancia mínima que queremos que mida, la cual va a ser una distancia de seguridad de 150cm, con estos datos podemos construir entonces:

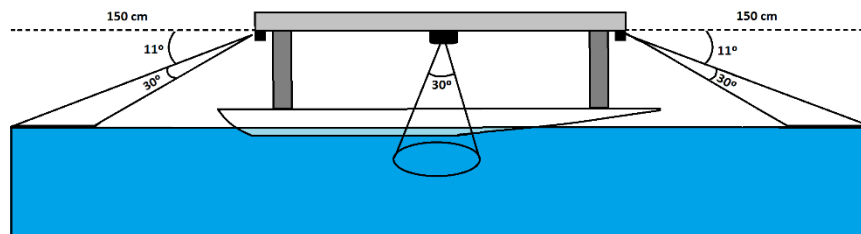


82. Triángulo de distancias

$$\tan x = \frac{\text{Altura del sensor}}{\text{Distancia horizontal a medir}} = \frac{30 \text{ cm}}{150 \text{ cm}} = \frac{1}{5}$$

$$x = \tan^{-1} \frac{1}{5} = 11,31^\circ$$

Según los cálculos, para conseguir medir el área deseada tendremos que colocar el soporte del sensor con un ángulo aproximadamente 11 grados, con lo que nos quedaría algo parecido a lo de la siguiente imagen:

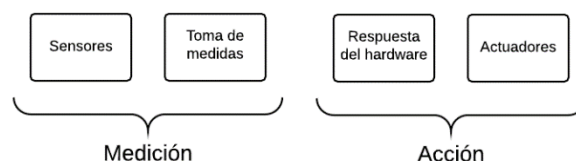


83. Campo de acción vertical de los sensores

Este fue el funcionamiento del prototipo en un primer momento, pero se encontró un gran problema. Arduino no es capaz de realizar dos tareas a la vez, lo que obligaba a hacer en primer lugar el barrido de los servos para la toma de medidas y después, el cálculo y ejecución de la respuesta. Esto convertía al sistema en uno muy lento, por lo que se tomó la decisión por el momento de sacrificar el barrido de los servos y tomar las medidas de los sensores ultrasonidos en estático, pudiendo así calcular las respuestas de acción de forma casi instantánea. Aunque se reduzca el campo de visión de esta forma no supone un gran inconveniente, ya que a una distancia prudente los sensores conservan un ángulo de visión aceptable. Esta modificación es temporal, ya que un trabajo futuro se plantea la sustitución del procesador de este sistema por uno más eficaz.

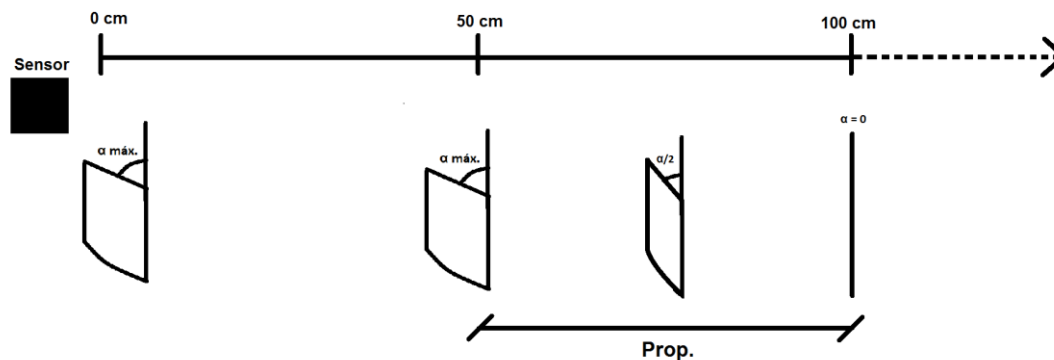
### Segundo método de programación (Reactivo)

Por lo tanto, se llegó a la programación de la balsa con un comportamiento completamente reactivo, basado en el desarrollo visto en *Introduction to AI robotics* (Murphy, 2000). En ella se distinguen 2 fases: medición y acción. Dentro de cada una de estas se encuentran las distintas partes de las que están compuestas. Se puede ver más gráficamente en el siguiente esquema:



84. Comportamiento reactivo

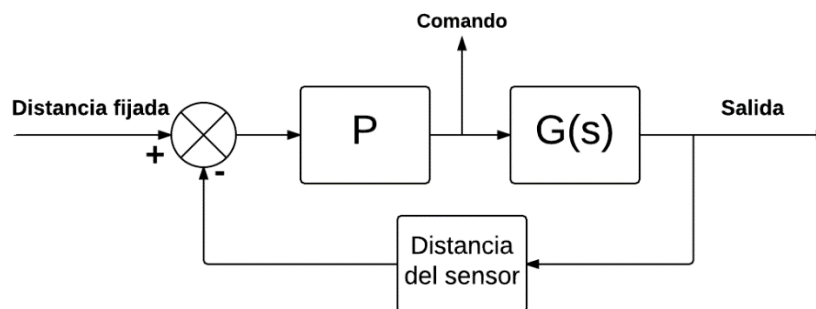
En resumen, el sistema toma unas medidas y ejecuta una respuesta. Esta estructura repetida de forma constante dota al prototipo de la capacidad de poder redirigir su trayectoria en cada momento. De esta forma se consigue un desplazamiento eficaz y seguro.



85. Giro proporcional del timón

El valor del ángulo del timón se va actualizando a cada ciclo, la balsa es capaz de ir tomando las medidas y de forma casi instantánea ajustar el giro del timón. Esto convierte al sistema en uno mucho más rápido comparado con el sistema explicado anteriormente.

Basándonos en la lógica del control proporcional y tomando  $G(s)$  como nuestro sistema, podemos esquematizar la situación de la siguiente forma:



86. Regulador proporcional

$$\text{Comando} = \text{Offset} + k * \text{error}$$

Siendo el error:

$$\text{Error} = \text{Distancia fijada} - \text{Distancia del sensor}$$

El sistema calcula una respuesta en función de la medida obtenida por el sensor, la distancia de seguridad prefijada y una constante de proporcionalidad previamente calculada que depende de los parámetros de la situación. De esta forma, el sistema calcula la variación del ángulo del timón que junto con un *Offset* expresa el ángulo final que tiene que tomar el timón en cada momento.

La lógica que sigue el algoritmo, según los parámetros escogidos, usada en la programación de este módulo podemos verla esquematizada en:

- Distancia medida  $> 1$  m. Aquí el sistema considera que no hay ningún peligro por lo que no ejecuta ninguna acción.
- $0.5\text{m} < \text{Distancia medida} < 1$  m. Aquí la regulación varía el giro del timón de manera proporcional según la distancia al obstáculo.
- Distancia medida  $< 0.5$  m. Aquí el ángulo del timón es máximo.

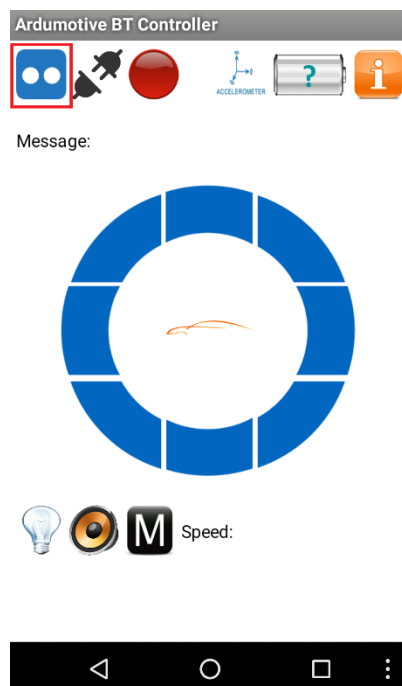
Las distancias usadas como ajuste en este caso son específicas para cada situación, en este caso se han ajustado a distancias pequeñas ya que las pruebas se realizarían en un entorno reducido. Para un entorno diferente sería conveniente recalcular estas ecuaciones de control.

En cuanto a la dirección de avance, el sistema avanza libremente hasta que se encuentra con un obstáculo por debajo de la distancia de seguridad, en ese momento comienza a retroceder mientras gira “huyendo” por el lado en el que la distancia lateral sea más prudente.



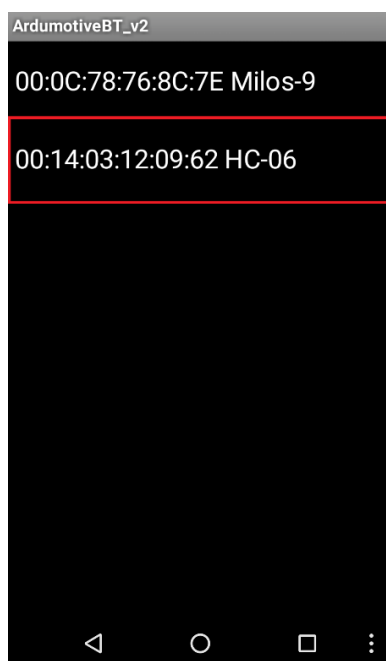
## 11. Descripción del funcionamiento

Al poner en marcha el sistema por defecto se inicia en modo manual. En este modo, podemos controlar el movimiento de la base con la aplicación móvil. Para ello, primero tenemos que conectar el bluetooth en nuestro dispositivo móvil, que lo haremos abriendo el desplegable para ver la lista de dispositivos disponibles:



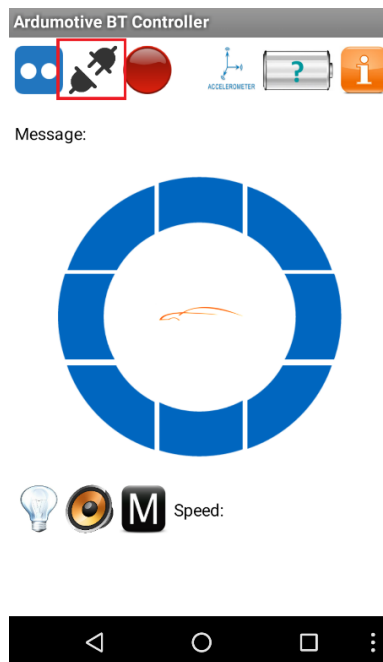
87. Ardumotive. Pantalla principal

Nos aparecerá algo como la imagen siguiente, debemos seleccionar HC-06, de esta forma nos conectaremos a la base:



88. Ardumotive. Dispositivos bluetooth

Ahora pulsaremos el botón “conectar” dentro de la aplicación para finalizar el proceso de conexión.

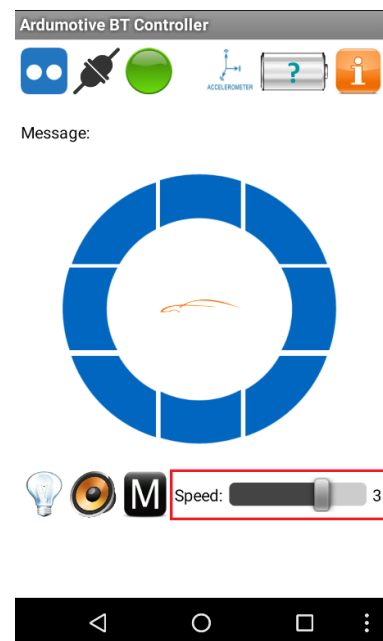


89. Ardumotive. Conexión

La luz roja en la aplicación parpadeará alternando entre rojo y verde hasta que la conexión se haya realizado con éxito. Una vez lograda, el led se fijará en verde.

Una vez llegado a este punto, podremos usar la ruleta central para elegir la dirección y el sentido de avance de la balsa.

Además, una vez iniciado el movimiento, podremos elegir la velocidad de avance con el selector de velocidad situado en la parte inferior derecha de la pantalla.



90. Ardumotive. Velocidad

Podremos usar una escala del 1 al 4 para hacer una división proporcional en la potencia suministrada al motor.

Así mismo, tenemos la opción de modo de navegación automático. En él, la base avanzará de forma autónoma hasta que se encuentre un obstáculo. En el caso de que se encuentre un obstáculo al lado izquierdo, la balsa girará hacia el lado derecho, y en caso de que se lo encuentre a su derecha, al contrario.

Al encontrarse un obstáculo de frente, la balsa retrocederá hasta una distancia prudente mientras gira para volver a avanzar hasta el lado en el que el obstáculo lateral esté más lejano.

La base navegará con este algoritmo hasta que haya algo que indique lo contrario, para hacer volver a la base hasta el punto deseado, basta con volver a poner el control en modo manual para reconducirla.

Toda la lógica que realiza la base está englobada en las siguientes funciones que han sido creadas específicamente para esta misión:

```
long MedirdistIzq();  
long MedirdistDel();  
long MedirdistTras();  
long MedirdistDcha();
```

#### *91. Funciones de medida*

Estas funciones son las encargadas de medir la distancia de cada sensor ultrasonidos, cada uno de ellas se encarga de medir el tiempo que tarda una señal de sonidos emitida en volver al emisor, en función del tiempo que trascurra en este proceso calcula la distancia al obstáculo.

Otras funciones usadas son las de dirección y sentido de movimiento:

```
void Avanzar();  
void Retroceder();  
void GirarIzq();  
void GirarDcha();  
void GirarIzqSuave();  
void GirarDchaSuave();  
void TimonRecto();  
void GirarServo(int angulo);
```

#### *92. Funciones dirección y movimiento*

Con estas funciones podemos girar el motor en sentido de avance, de retroceso, o cambiar el ángulo del timón, ya sea con giro suave (15°), giro completo (35°) o volverlo a la posición neutral (0°). También hay una función de giro en la que el ángulo se pasa por un parámetro. Esta función se usa para la regulación automática del servo en modo evasión de obstáculos.

Por último, tenemos otras funciones programadas:

```
void MedirBateria();  
void MedirVelocidad();  
void Brujula();
```

#### *93. Funciones varias*

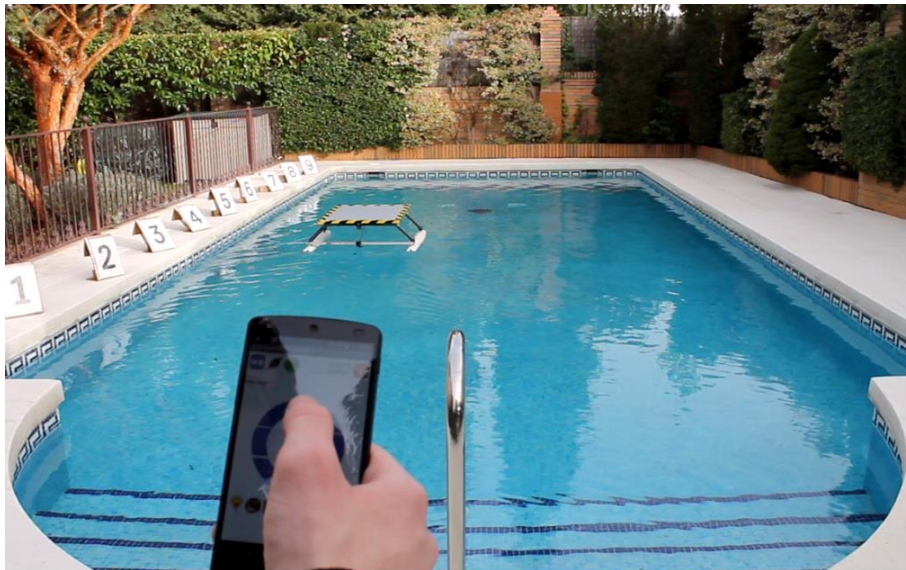
Con estas funciones podemos: medir el voltaje de la batería, nos aparecerá en el registro de datos además, se encarga de emitir una señal sonora cuando es necesario recargar la batería. La siguiente función calcula la velocidad de avance de la balsa en función de las aceleraciones medidas por la IMU. La cuál también participa en la última función, que es la encargada de mostrar la orientación de la balsa en cada momento.

## 12. Pruebas y ensayos

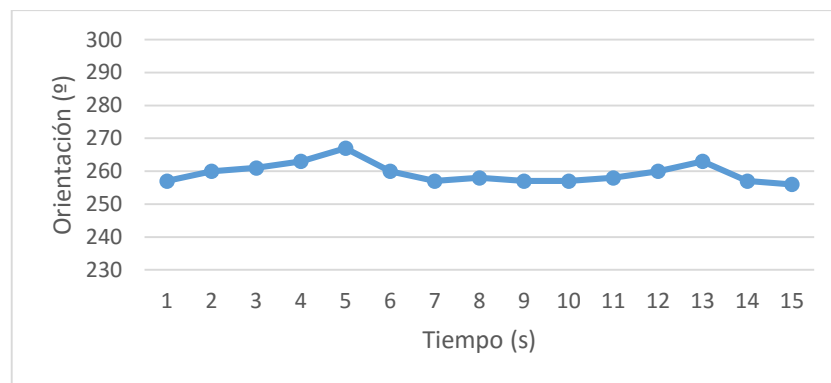
Para demostrar el correcto funcionamiento del producto fabricado se han realizado 3 pruebas distintas. Las dos primeras en modo manual en la que manejará con el envío de los comandos a través del dispositivo móvil. La primera consiste en la realización de una línea recta para ver la capacidad de avance de la base y la segunda, la realización de una *U* para comprobar la efectividad del giro. Existe una tercera prueba para demostrar el funcionamiento del modo evasión de obstáculos en la cual la balsa trazará su propia trayectoria en función de las instrucciones que le han sido programadas. Se aprecia como el prototipo cambia su rumbo al detectar un objeto que pueda interferir en su trayectoria y se dirige a un nuevo camino libre de obstáculos.

Más abajo se encuentran las gráficas obtenidas con la representación de los datos obtenidos durante la ejecución de las pruebas acompañadas con una captura de la prueba en sí. No obstante, se mostrará un vídeo con el desarrollo de la prueba completa.

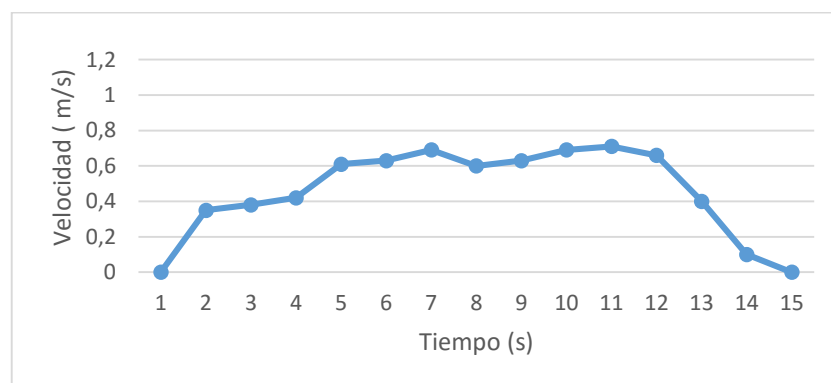
## Línea recta



94. Prueba línea recta



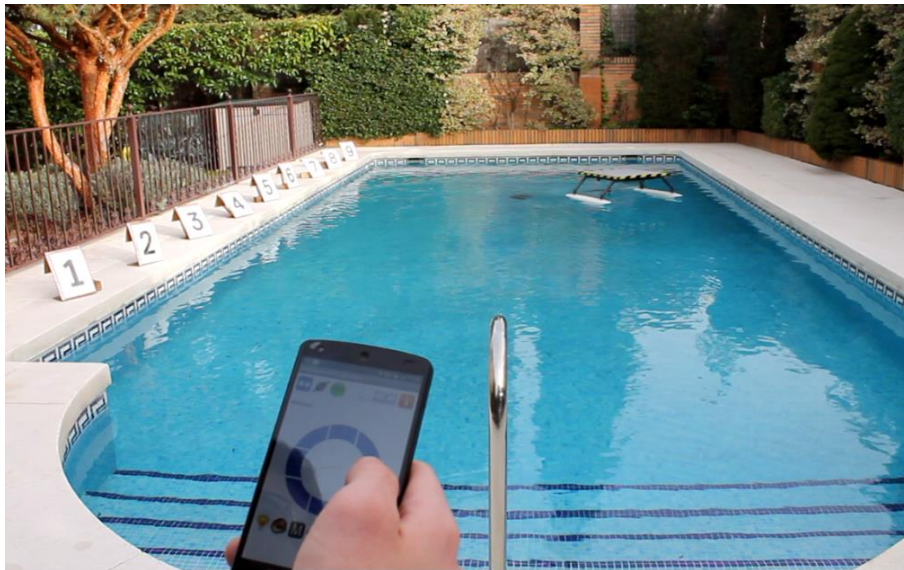
95. Gráfico orientación. Línea recta



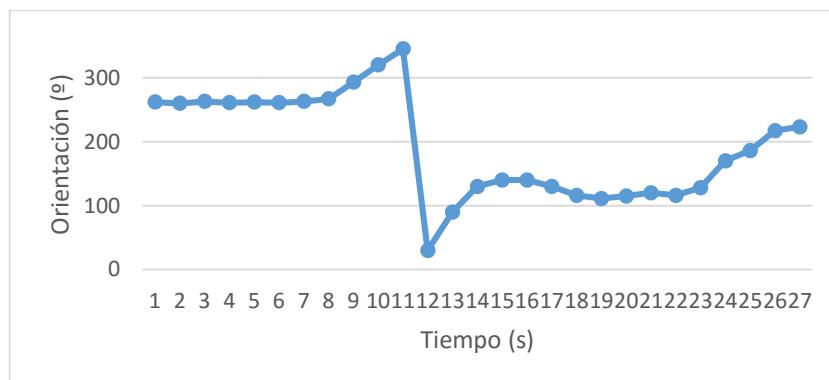
96. Gráfico velocidad. Línea recta

Se puede observar tres zonas en la gráfica de velocidad. Una primera de aceleración, una intermedia donde la velocidad es constante y una última de frenado. En la parte de intermedia podemos ver ligeras variaciones de velocidad, esto se produce al realizar las correcciones de trayectoria que podemos ver reflejados en la orientación de la primera gráfica.

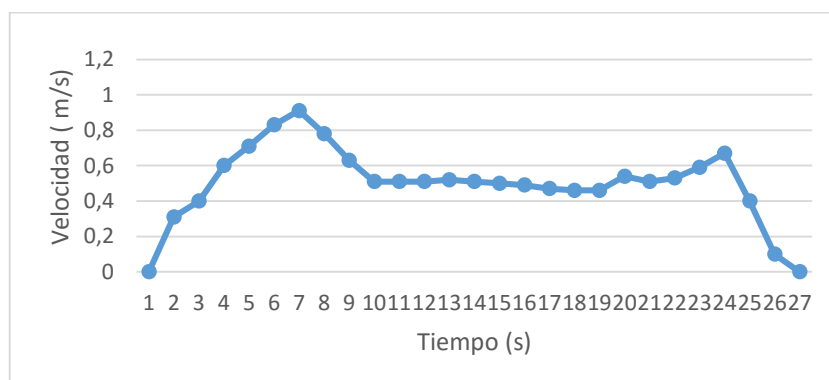
## Prueba U



97. Prueba U



98. Gráfico orientación. U



99. Gráfico velocidad. U

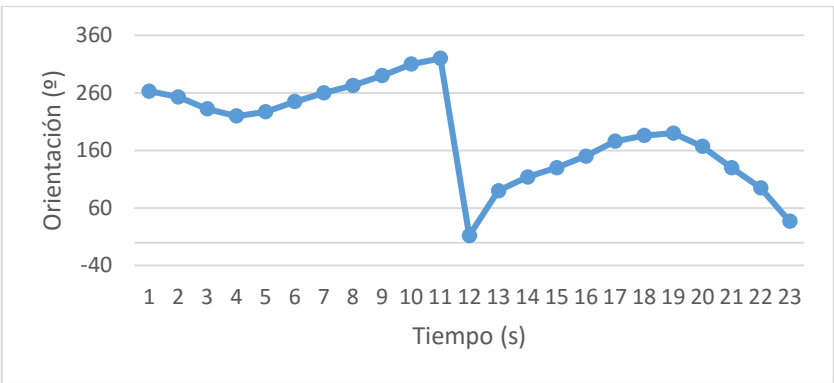
En esta prueba vemos un caso parecido al anterior, salvo que a mitad del recorrido se produce un cambio de orientación debido a un cambio de sentido de la trayectoria del prototipo. También se encuentran en esta prueba la zona de aceleración y frenado en la gráfica de velocidad y una zona intermedia más alargada y estable que en la prueba anterior, en la que vemos que la velocidad se mantiene casi constante durante el giro.



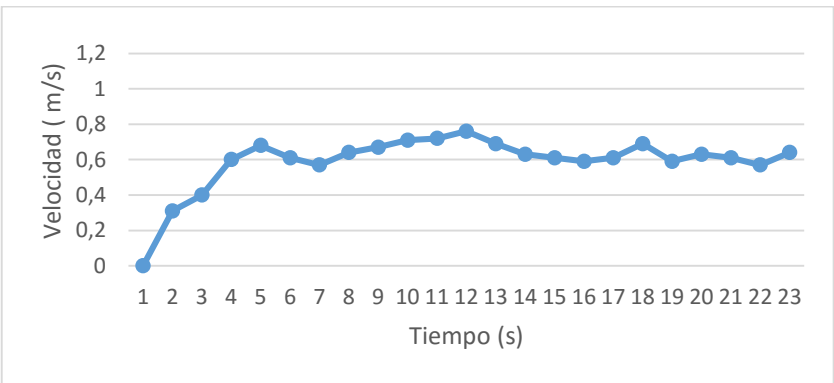
Evasión de obstáculos



100. Prueba evasión de obstáculos



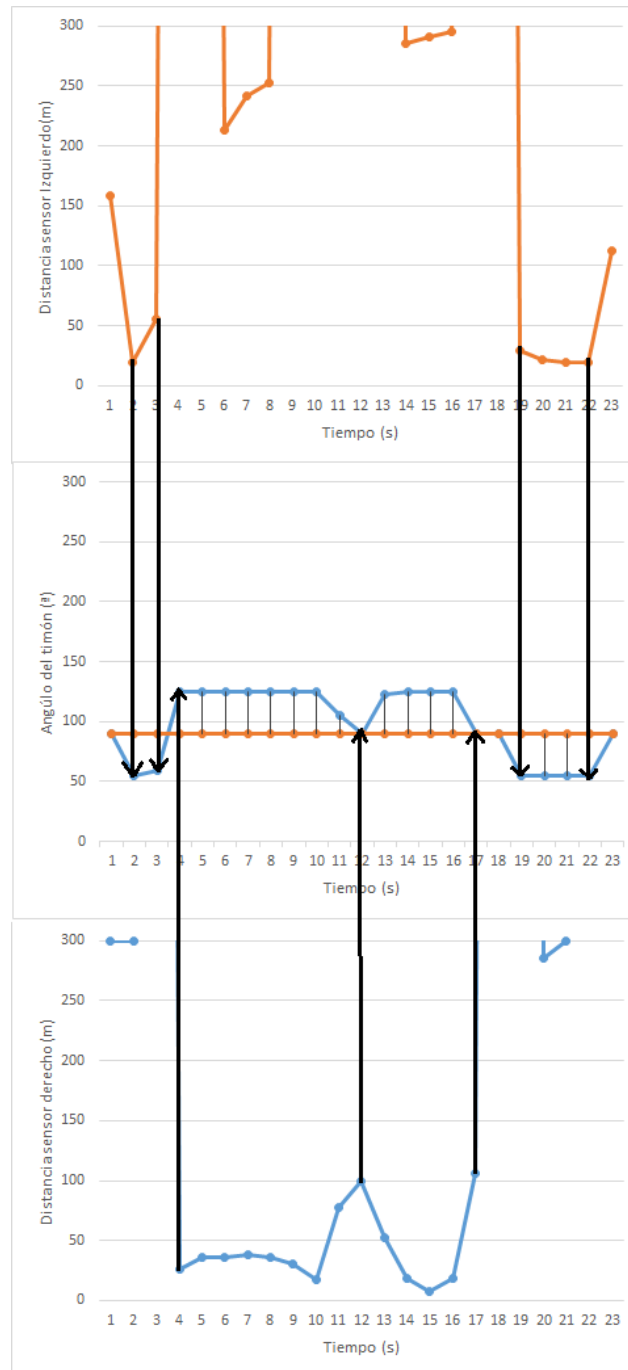
101. Gráfica orientación. Evasión de obstáculos



102. Gráfica velocidad. Evasión de obstáculos

En este caso, a diferencia de los anteriores es la balsa la que gestiona sus propios cambios de trayectorias. Se puede intuir desde la primera gráfica que se interpondrá un obstáculo en el camino en el segundo 12 y 19, que son donde la orientación sufre un cambio significativo y una leve disminución de la velocidad.

A diferencia de las pruebas anteriores, en esta contamos también con unas gráficas en las que se representa en primer lugar la distancia medida por el sensor izquierdo, después una gráfica en la que se representa el ángulo del timón con una línea incluida con la posición neutral del timón (90°). Y por último, una gráfica con las distancias medidas por el sensor derecho.



103. Gráficas distancia sensores y timón. Evasión de obstáculos

Con estas gráficas podemos observar el comportamiento reactivo del sistema. Vemos que justo cuando se detecta un obstáculo por un lado por debajo de 1 metro, el timón reacciona girando en un sentido hasta llegar a su máximo en 50 cm. En el sentido contrario cuando la aproximación se produce por el lado opuesto. Además, podemos ver que este cambio se hace de forma proporcional.

En las gráficas de las mediciones de los sensores podemos ver que algunos valores se salen de la tabla, esto ocurre cuando el sensor no detecta ningún obstáculo cercano. La medición de los sensores ultrasonidos se produce mediante la medición del tiempo que tarda en volver una onda, si esta nunca vuelve, no se podrá realizar dicha medición por lo que el sensor dará un valor anormalmente elevado.

## 13. Problemas y limitaciones encontrados

Uno de los problemas encontrados durante el desarrollo de este proyecto es la limitación que proporciona Arduino. Este solo es capaz de realizar una tarea a la vez, es decir, tiene una ejecución secuencial del código, por lo tanto, no es capaz por ejemplo de hacer un barrido con los servos mientras toma las distancias de los sensores ultrasonidos y ejecuta las acciones correspondientes. Por esto, para el futuro es conveniente sustituir este Arduino por procesador más avanzando como Raspberry Pi o Intel Edison, que son capaces de gestionar diferentes procesos simultáneamente.

Otro problema encontrado es con el controlador del motor (L298N), ya que debido a su lógica de funcionamiento constituye una respuesta de reacción más lenta. Este controlador tiene dos partes, un divisor de tensión que toma valores en el rango 0-255, y una combinación de dos bits que es la que marca el sentido de giro. Por lo tanto, el comportamiento reactivo de este componente no es tan rápido como puede ser la respuesta del servo del timón no es algo inmediato. Como, en principio, no es tan necesaria la inmediata respuesta en la velocidad y el sentido de avance, se ha decidido no investigar esta parte. Dejando pendiente la elaboración de un método más efectivo para este tipo de dispositivos o en caso de no ser posible, la sustitución del mismo.

Además, también ha sido una fuente de problemas la oscilación de las mediciones realizadas por el acelerómetro, ya que este componente en una misma situación proporciona medidas ligeramente diferentes. Por ello, se aisló este componente de la caja de electrónica, pero aun así, existen pequeñas variaciones en la medición. Esto conlleva pequeñas variaciones en la medición de la aceleración, y por lo tanto un pequeño error en el cálculo de la velocidad, lo que hace a consecuencia que nuestro sistema sea menos preciso. Por este motivo, sería conveniente para el futuro sustituir este componente por uno más eficaz o bien, la realización de un filtro Kalman para absorber estos errores de medida.

## 14. Posibles extras y mejoras futuras

A lo largo de la creación de este proyecto han ido ocurriendo diversas modificaciones y/o mejoras posibles que poder aplicar, pero ya sea por falta de tiempo o porque se escapaban de los objetivos de este trabajo, no se han llegado a implantar. No obstante, se comentarán aquí para futuras referencias o procesos.

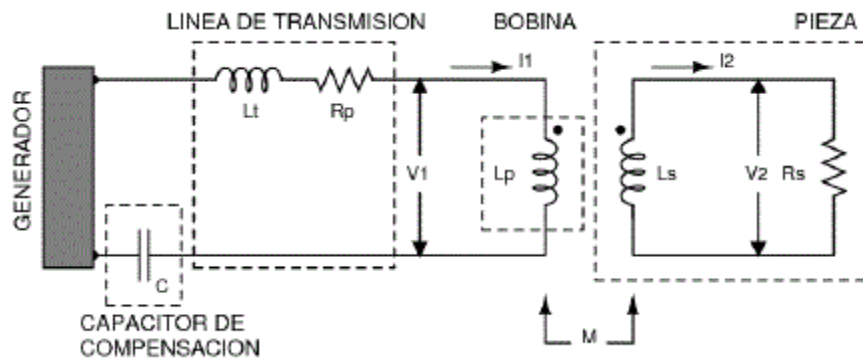
El principal problema en este tipo de proyectos es la limitada autonomía que ofrecen las baterías, y en nuestra misión es necesaria una extensa capacidad de actuación, ya que precisamente lo que queremos es servir como apoyo al vehículo aéreo. Para solucionar este problema podrían implantarse placas solares para la recarga continua de las baterías.



*104. Placa Fotovoltaica (Amazon, 2016)*

Existen a disposición del consumidor placas solares de dimensiones reducidas, las cuales permitirían su instalación en este trabajo. Además de su coste razonable, estas placas proporcionan la suficiente potencia como para cargar las baterías que usamos, por lo que en condiciones meteorológicas aceptables no supondría ningún inconveniente el uso continuado de este elemento de apoyo.

La base no es la única con problemas de autonomía, ya que el dron usa baterías de características similares y, además, con un mayor consumo, por lo que también la duración de las baterías supone un problema en este caso. Al fin y al cabo, esta base es un apoyo en la misión de reconocimiento de un dron, por lo que no sería extraño proveer a la base con un sistema de carga inalámbrica para la fuente de alimentación de este. Para ello, se puede usar un circuito por carga por inducción eléctrica como la de la figura de abajo.



105. Alcance del Sharp 1 (Scielo, 2016)

Poniendo uno de estas partes del circuito en la base, cerca de la superficie de aterrizaje, y la otra conectada a la batería del dron, permitiría la carga inalámbrica una vez que el dron haya alcanzado la superficie de aterrizaje. Como consecuencia este podría recuperar su fuente de alimentación y volver al vuelo una vez acabado el proceso.

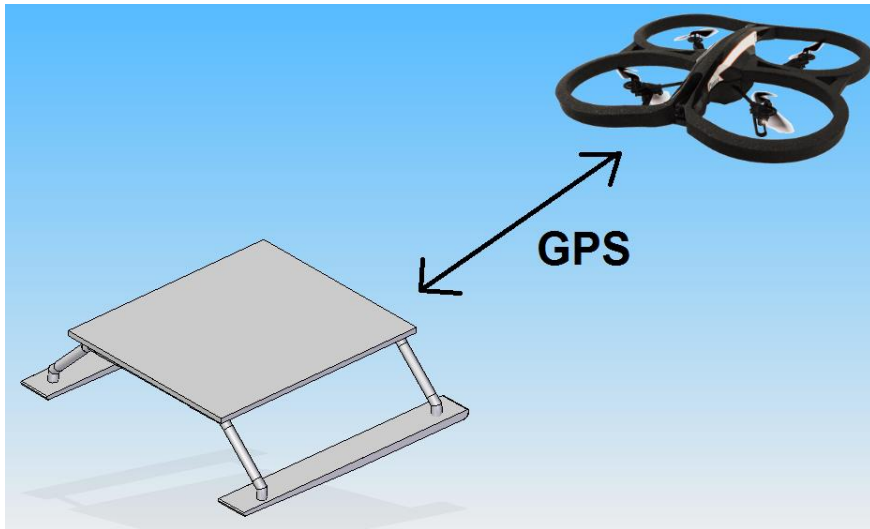
Otro extra con posibilidad de ser incorporado es una cobertura de goma para la superficie de aterrizaje.



106. Estera con puntos antideslizantes (Mercamania, 2016)

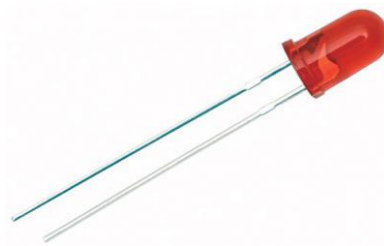
El objetivo de esta modificación es proveer a la base con una superficie antideslizante, para que, en caso de que existan inestabilidades en el medio acuático, el dron no resbale y caiga al agua, ya que una superficie gomosa en contacto con el dron sujetaría a este en caso de turbulencias.

Una mejora muy importante a tener en cuenta es un sistema de navegación. Con esto la base sería capaz de navegar de forma autónoma hasta la ubicación establecida usando el GPS para conocer el punto de destino y la brújula para la orientación. Este aspecto puede ser muy útil si al lograr una comunicación directa entre el dron y la balsa, aquel le envía su ubicación a esta y mediante un algoritmo calcula la ruta a seguir. De esta forma, la balsa seguiría constantemente al dron para estar cerca de él para hacerle de sistema de apoyo.



107. Comunicación Dron-Balsa

Un último detalle que puede ser incluido, ya sea por estética o por utilidad, sería una delimitación del perímetro de la balsa mediante led rojos.



108. LED Rojo (Stereon, 2016)

Programándolos para que estos parpadeen puede servir a modo de localización en caso de visibilidad reducida o ambientes nocturnos. Esto puede ser útil a la hora de localizar la base por parte de su piloto o incluso desde el propio sistema de visión del dron que busca a esta para realizar su aterrizaje.



## 15. Conclusiones

La realización de este trabajo ha supuesto todo un reto, ya que ha sido necesaria la utilización de conocimientos de muchas áreas. Además, ha sido fuente de numerosos imprevistos y contratiempos que me han llevado a la búsqueda de soluciones alternativas o adaptaciones. Componentes que no han respondido como se esperaba o incluso algunos que han sufrido daños durante la fase de pruebas y han tenido que ser rápidamente sustituidos. Todo ello ha conllevado el desarrollo de una gran capacidad resolutive.

También, ha sido todo un descubrimiento en el terreno de los sistemas de control, ya que el conocimiento extraído de los numerosos artículos y libros consultados sobre el tema me han hecho adquirir una visión más global del mundo de la electrónica y la automática. La aplicación de dichos conocimientos a este trabajo ha hecho posible el correcto funcionamiento del mismo e incluso el planteamiento de mejoras futuras que dotarán a este prototipo de características aún más interesantes.

Los resultados obtenidos han sido los esperados en gran parte, no obstante, hay algunos aspectos que aún quedan por desarrollar o mejorar. Ya que la mejora de este proyecto es aún algo en lo que queda mucho por descubrir.

En mi opinión, este proyecto constituye un avance notable en su área de investigación, ya que es algo novedoso y de gran utilidad a el que se le puede aún sacar mucho más provecho con la implantación de nuevos usos y variantes adaptadas a las distintas situaciones en las que se requiera un sistema de estas características.

## 16. Bibliografía

- ADAFRUIT (2016). Programming and Calibration | ADXL345 Digital Accelerometer | Adafruit Learning System. [Online] Obtenido de: <https://learn.adafruit.com/adxl345-digital-accelerometer/programming> [Fecha de acceso: 16 Feb. 2016].
- ALVES, J. y DIAS, J. (2003). "Design and control of a spherical mobile robot". Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering, 217(6), pp.457-467.
- ARDUMOTIVE (2015). *Arduino Bluetooth RC Car (Android Controlled)*. [Online] Available at: <http://www.ardumotive.com/bluetooth-rc-car.html>. [Fecha de acceso: 7 Dic.2015].
- AUTODESK, (2015). *Calculating the Coefficient of Drag - Urbee Example | Sustainability Workshop*. [Online] Disponible en: <http://sustainabilityworkshop.autodesk.com/products/calculating-coefficient-drag-urbee-example>. [Fecha de acceso: 03 Oct. 2015]
- BARBUS, E. (2014). *El cajón de Arduino: Tutorial: sensor ultrasonidos HC-SR04*. [Online] Elcajondeardu.blogspot.com.es. Obtenido de: <http://elcajondeardu.blogspot.com.es/2014/03/tutorial-sensor-ultrasonidos-hc-sr04.html>. [Fecha de acceso: 16 Oct.2015].
- DIYMAKERS, (2014). *Arduino + Bluetooth*. [Online] Obtenido de: <http://diymakers.es/arduino-bluetooth/>. [Fecha de acceso: 23 Dic.2015].
- DONWEB, (2014). *Los 14 usos de drones que seguro no conocías | Agencia de Noticias Tecnológicas DonWeb*. [Online] Obtenido de: <http://agencia.donweb.com/los-14-usos-de-drones-que-seguro-no-conocias/>. [Fecha de acceso: 1 Dic.2015].
- EDUCACHIP, (2014). *Cómo Leer Y Escribir Datos En La Tarjeta SD De Arduino*. [Online] Obtenido de: <http://www.educachip.com/como-leer-y-escribir-datos-en-la-tarjeta-sd-de-arduino/> [Fecha de acceso: 12 Feb. 2016].
- ELECTROENSAIMADA, (2015). *Tutorial Arduino: Servomotor*. [Online] Obtenido de: <http://www.electroensaimada.com/servomotor.html>. [Fecha de acceso: 05 Ene.2016].
- GERR, D. (1989). *Propeller handbook*. Camden, Me.: International Marine Pub. Co. (2001).
- INCROPERA, F., DEWITT, D. y CRUZ, R. (1999). *Fundamentos de transferencia de calor*. México: Prentice-Hall (2009).
- MH-AEROTOOLS, (2015). *How a Propeller Works*. [Online] Obtenido de: <http://www.mh-aerotoools.de/airfoils/propuls4.htm>. [Fecha de acceso: 13 Nov.2015].
- MURPHY, R. (2000). *Introduction to AI robotics*. Cambridge, Mass.: MIT Press (2001).
- NASASPACEFLIGHT, (2016). *SpaceX Augments and Upgrades Drone Ship Armada | NASASpaceFlight.com*. [Online] Obtenido de:

- <http://www.nasaspaceflight.com/2015/06/spacex-augments-upgrades-drone-ship-armada/> [Fecha de acceso: 9 Feb. 2016].
- NAUTICA&YATES, (2015). *Cinco drones resistentes al agua que pueden grabar imágenes submarinas* Aeronort UAV Services, (2016). *Medio ambiente - Aeronort UAV Services*. [Online] Obtenido de: <http://www.aeronort.com/medio-ambiente/>. [Fecha de acceso: 4 Ene.2016].
- NAUTICA&YATES. [Online] Obtenido de: <http://www.nauticayyates.com/cinco-drones-resistentes-al-agua-que-pueden-grabar-imagenes-submarinas-waterproof/>. [Fecha de acceso: 7 Ene.2016].
- NAYLAMPMECHATRONICS, (2015). *Tutorial de Uso del Módulo L298N*. [Online] Obtenido de: [http://www.naylampmechatronics.com/blog/11\\_Tutorial-de-Uso-del-M%C3%B3dulo-L298N.html](http://www.naylampmechatronics.com/blog/11_Tutorial-de-Uso-del-M%C3%B3dulo-L298N.html). [Fecha de acceso: 1 Dic.2015].
- LATABERNADELPUERTO (2016). *Angulo de giro de un timon 45 grados o más - La Taberna del Puerto*. [Online] Obtenido de: <http://foro.latabernadelpuerto.com/showthread.php?t=76795> [Fecha de acceso: 17 Feb. 2016].
- ORE, J., ELBAUM, S., BURGIN, A. AND DETWEILER, C. (2015). "Autonomous Aerial Water Sampling". *Journal of Field Robotics*, 32(8), pp.1095-1113.
- ORWIG, J. (2016). *Elon Musk Just Unveiled A Game-Changing Ocean Landing Pad For His Reusable Rockets*. [Online] Business Insider. Obtenido de: <http://www.businessinsider.com/elon-musk-rockets-that-land-at-sea-2014->. [Fecha de acceso: 17 Ene.2016].
- OUNAE, (2014). *Los 10 usos más sorprendentes de los drones*. [Online] Obtenido de: <http://ounae.com/aplicaciones-usos-drones/>. [Fecha de acceso: 1 Dic.2015].
- PARROT, (2015). *AR.Drone 2.0. Parrot new wi-fi quadricopter - AR.Drone.com - HD Camera - Civil drone - Parrot*. [Online] Obtenido de: <http://ardrone2.parrot.com/>. [Fecha de acceso: 17 Nov.2015].
- PINTO, E., MARQUES, F., MENDONCA, R., LOURENCO, A., SANTANA, P. AND BARATA, J. (2014). An autonomous surface-aerial marsupial robotic team for riverine environmental monitoring: Benefiting from coordinated aerial, underwater, and surface level perception. *2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014)*.
- REES, C. (2015). *The Mariner 600 UAV-ROV | Unmanned Systems Technology*. [Online] Unmanned Systems Technology. Obtenido de: <http://www.unmannedsystemstechnology.com/2015/04/search-systems-launches-aquatic-uav-rov-crossover/mariner-600-uav/>. [Fecha de acceso: 17 Ene.2016].

- ROBOLOGS, (2014). *Tutorial de Arduino y MPU-6050 – robologs*. [Online] Obtenido de: <http://robologs.net/2014/10/15/tutorial-de-arduino-y-mpu-6050/> [Fecha de acceso: 12 Feb. 2016].
- SPARKFUN (2016). ADXL345 Quickstart Guide - SparkFun Electronics. [Online] Obtenido de: <https://www.sparkfun.com/tutorials/240> [Fecha de acceso: 15 Feb. 2016].
- STAFF, G. (2015). *Researchers to Test Water Drone for Bridge Inspections*. [Online] GPS World. Obtenido de: <http://gpsworld.com/researchers-to-test-water-drone-for-bridge-inspections/>. [Fecha de acceso: 17 Ene.2015].
- YAVUZ, H. (2007). "An integrated approach to the conceptual design and development of an intelligent autonomous mobile robot". *Science Direct, Robotics and Autonomous System* (55). Págs 498-512.

## ANEXO: CÓDIGO ARDUINO

```
#include <Servo.h>
#include <SoftwareSerial.h>
#include <SPI.h>
#include <SD.h>
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_ADXL345_U.h>
#include <HMC5883L.h>
/*****Definición de variables*****/
Adafruit_ADXL345_Unified accel = Adafruit_ADXL345_Unified(12345);
//Variables para los sensores de distancia
long distanciazq;
long tiempolzq;
long distanciaDel;
long tiempoDel;
long distanciaTras;
long tiempoTras;
long distanciaDcha;
long tiempoDcha;
long menordistlzq;
long menordistDel;
long menordistTras;
long menordistDcha;
//Definición de pines
int IN3 = 26; // Con IN3 y IN4 elegimos el sentido de giro del motor
int IN4 = 24;
int ENB = 10; // ENB activa o desactiva el motor
SoftwareSerial BT(12,13);
//Variables y constantes para la medición de batería
unsigned long Milisactuales; //Variables para hacer posible la medición a intervalos
long Milisanteriores = -1000*10;
long intervalo = 1000*10;
const float BateriaMax = 8; //Voltaje máximo de la batería
int PorVolt;
float voltaje = 0.0;
int nivel;
int zumbador = 46; //Señal de alerta de la batería baja
//Definición de variables varias
char estado;
int modo=0;
int angulo = 90;
int velocidad = 255;
long DistSeg = 50; //Distancia de seguridad para la que la base rectifica la trayectoria
File Log;
unsigned long Milisactuale; //Variables para hacer posible la medición a intervalos
```

```

long Milisanteriore = -100;
long interval = 100;
//Creación del servo del timón
Servo ServoTimon;
//Creación de los servos de barrido
Servo servo;
Servo servo2;
Servo servo3;
Servo servo4;
int pos = 90;// Indicador de posición para el servo de barrido
//Variables para la medición de velocidad
float t0=0;
float t1=0;
float a0=0;
float a1=0;
float v=0;
//Variables brújula
HMC5883L compass;
int error = 0;
/*****Definición de pines y
comunicaciones*****/
void setup() {
//Definición de la velocidad de comunicación
Serial.begin(9600);
BT.begin(9600);
//Definición del modo de los pines
//Pines de los sensores ultrasonidos
//Sensor izquierdo
pinMode(35, OUTPUT);
pinMode(34, INPUT);
//Sensor delantero
pinMode(37, OUTPUT);
pinMode(36, INPUT);
//Sensor trasero
pinMode(7, OUTPUT);
pinMode(6, INPUT);
//Sensor derecho
pinMode(9, OUTPUT);
pinMode(8, INPUT);
//Pines del controlador del motor
pinMode (ENB, OUTPUT);
pinMode (IN3, OUTPUT);
pinMode (IN4, OUTPUT);
//Pin del servo del timón
ServoTimon.attach(42);
//Pines de los servos de barrido
servo.attach(30);
servo2.attach(31);
servo3.attach(32);
servo4.attach(33);

```



```

//Pin del zumbador
pinMode(zumbador, OUTPUT);
/*****Iniciación Módulo SD*****/
Serial.print("Iniciando tarjeta SD...");
pinMode(53,OUTPUT);
if (!SD.begin(53)) {
Serial.println("Iniciación SD fallida!");
return;
}
Serial.println("Iniciación completa.");
Log = SD.open("datos.txt", FILE_WRITE); //Abrir archivo
/*****Acelerómetro*****/
if(!accel.begin())
{
/* There was a problem detecting the ADXL345 ... check your connections */
Serial.println("Ooops, se ha detectado el ADXL345!");
while(1);
}
sensor_t sensor;
accel.setRange(ADXL345_RANGE_16_G);/* Seleccionar rango de medición */
/*****Brújula*****/
**/
Wire.begin();
compass = HMC5883L();
error = compass.SetScale(1.3); // Seleccionar escala
if(error != 0)
Serial.println(compass.GetErrorText(error));
error = compass.SetMeasurementMode(Measurement_Continuous);
if(error != 0)
Serial.println(compass.GetErrorText(error));
}
/*****Ejecución del programa principal*****/
void loop() {
Serial.println("Sistema en marcha. Iniciado en modo manual");
Log.println("Sistema en marcha. Iniciado en modo manual");
/*****Sensores ultrasonidos*****/
//Funciones para medir la distancia en el sensor izquierdo, delantero, trasero
y derecho respectivamente.
if(modos==0){
MedirdistIzq();
MedirdistDel();
MedirdistTras();
MedirdistDcha();
Serial.println(" ");
Milisactuale = millis(); //Almacena los milisegundos actuales
if(Milisactuales - (Milisanteriore) > (interval)) { //Comprueba si ha pasado
ya el tiempo de intervalo suficiente
Milisanteriore = Milisactuale; //
Serial.println("Grabando");

```

```

Log.print("t = ");Log.println(Milisactuale);
Brujula();
MedirVelocidad();
Serial.println(" ");
}
Serial.println(" ");
}
//*****Servo timon modo automático*****//
while (modo==1){
Serial.println("Inicio en modo automático");
Log.println("Inicio en modo automático");
MedirdistIzq();
MedirdistDel();
MedirdistTras();
MedirdistDcha();
Serial.println(" ");
if(distanciaDcha<distanciaIzq){
angulo = 125 - 0.7*(distanciaDcha-DistSeg);
if(angulo<90) angulo=90;
if(angulo>125) angulo=125;
GirarServo(angulo);
}
if(distanciaIzq<distanciaDcha){
angulo = 55 + 0.7*(distanciaIzq-DistSeg);
if(angulo<55) angulo=55;
if(angulo>90) angulo=90;
GirarServo(angulo);
}
if(distanciaDel<50){
Retroceder();
if (menordistIzq<menordistDcha){
GirarDcha();
}else{
GirarIzq();
}
}
if(distanciaDel>50){
Avanzar();
}
estado = BT.read();
if (estado == 'W') {
Serial.println("Cambio de modo");
modo=0;
analogWrite(ENB,0);
}
Milisactuale = millis(); //Almacena los milisegundos actuales
if(Milisactuales - (Milisanteriore) > (interval)) { //Comprueba si ha pasado
ya el tiempo de intervalo suficiente
Milisanteriore = Milisactuale; //
Serial.println("Grabando");
}

```

```

Log.print("t = ");Log.println(Milisactuale);
Brujula();
MedirVelocidad();
Log.print("Angulo timón: ");Log.println(angulo);
Log.print("Distancia Dcha "); Log.print(distanciaDcha); Log.println(" cm");
Log.print("Distancia Det "); Log.print(distanciaTras); Log.println(" cm");
Log.print("Distancia Izq "); Log.print(distanciaIzq); Log.println(" cm");
Log.print("Distancia Del "); Log.print(distanciaDel); Log.println(" cm");
}
Log.flush(); //Guardado del registro de datos
Log.println(" ");
MedirBateria();
delay(100);
}
//*****Bluetooth*****//
estado = BT.read();
//Cambia la velocidad de 0 a 4 haciendo un divisor de tensión escribiendo de 0
a 255 en una salida PWM
if (estado == '0'){
velocidad=0;}
else if (estado == '1'){
velocidad=100;}
else if (estado == '2'){
velocidad=180;}
else if (estado == '3'){
velocidad=200;}
else if (estado == '4'){
velocidad=255;}
/*****Hacia delante*****/
//En el estado F movemos el motor para desplazar la base hacia delante
if (estado == 'F') {
Avanzar();
TimonRecto();
}
/*****Hacia delante e izquierda*****/
//El estado G es igual que el anterior pero además gira a la izquierda
else if (estado == 'G') {
Avanzar();
GirarIzqSuave();
}
/*****Hacia delante y derecha*****/
//Estado T. Igual que el anterior pero hacia la derecha
else if (estado == 'I') {
Avanzar();
GirarDchaSuave();
}
/*****Hacia atrás*****/
//En el estado B el motor gira en sentido inverso
else if (estado == 'B') {
Retroceder();
}

```

```

TimonRecto();
}
/*****Hacia atrás e izquierda*****/
//En el estado H hacemos lo mismo que en anterior pero girando además a la
derecha
else if (estado == 'H') {
Retroceder();
GirarlzqSuave();
}
/*****Hacia detrás y derecha*****/
//J.Igual que el anterior pero a la derecha
else if (estado == 'J') {
Retroceder();
GirarDchaSuave();
}
/*****Izquierda*****/
//Únicamente giro a la izquierda
else if (estado == 'L') {
Girarlzq();
Avanzar();
}
/*****Derecha*****/
//Únicamente giro a la derecha
else if (estado == 'R') {
GirarDcha();
Avanzar();
}
/*****Parada*****/
//En este estado se para el motor
else if (estado == 'S'){
analogWrite(ENB,0);
}
/*****Cambio de modo*****/
else if (estado == 'W') {
modo=0;
analogWrite(ENB,0);
} else if (estado == 'V'){
modo=1;
}
MedirBateria();
//Para el siguiente bucle
Serial.println(" ");
Log.flush(); //Guardado del registro de datos
delay(100);
}
long Medirdistlzq(){
digitalWrite(35, LOW); //Estabilización de sensores
delayMicroseconds(5);
digitalWrite(35, HIGH); // Envío del pulso ultrasónico
delayMicroseconds(10);
}

```

```

tiempolzq = pulseIn(34, HIGH); // Función para medir la longitud del pulso
entrante lzq.
distancialzq = int(0.017 * tiempolzq); // Fórmula para calcular la distancia
lzq obteniendo un valor entero
Serial.print("Distancia lzq "); Serial.print(distancialzq); Serial.println("
cm");
return distancialzq;
}
long MedirdistDel(){
digitalWrite(37, LOW);
delayMicroseconds(5);
digitalWrite(37, HIGH);
delayMicroseconds(10);
tiempoDel = pulseIn(36, HIGH);
distanciaDel = int(0.017 * tiempoDel);
Serial.print("Distancia Del "); Serial.print(distanciaDel); Serial.println("
cm");
return distanciaDel;
}
long MedirdistTras(){
digitalWrite(7, LOW);
delayMicroseconds(5);
digitalWrite(7, HIGH);
delayMicroseconds(10);
tiempoTras = pulseIn(6, HIGH);
distanciaTras = int(0.017 * tiempoTras);
Serial.print("Distancia Det "); Serial.print(distanciaTras); Serial.println("
cm");
return distanciaTras;
}
long MedirdistDcha(){
digitalWrite(9, LOW);
delayMicroseconds(5);
digitalWrite(9, HIGH);
delayMicroseconds(10);
tiempoDcha = pulseIn(8, HIGH);
distanciaDcha = int(0.017 * tiempoDcha);
Serial.print("Distancia Dcha "); Serial.print(distanciaDcha); Serial.println("
cm");
return distanciaDcha;
}
void Avanzar(){
digitalWrite (IN3, HIGH);
digitalWrite (IN4, LOW);
analogWrite(ENB, velocidad);
Serial.println("Avanzar");
Log.println("Avanzar");
}
void AvanzarProp(int velocidad){
digitalWrite (IN3, HIGH);

```

```

digitalWrite (IN4, LOW);
analogWrite(ENB,velocidad);
Serial.println("Avanzar");
Log.println("Avanzar");
}
void Retroceder(){
digitalWrite (IN3, LOW);
digitalWrite (IN4, HIGH);
analogWrite(ENB,velocidad);
Serial.println("Retroceder");
Log.println("Retroceder");
}
void MedirBateria(){
/*****Medidor de batería*****/
Milisactuales = millis(); //Almacena los milisegundos actuales
if(Milisactuales - (Milisanteriores) > (intervalo)) { //Comprueba si ha
pasado ya el tiempo de intervalo suficiente
Milisanteriores = Milisactuales; //Guarda el almacenaje como anteriores
para la siguiente pasada
voltaje = (analogRead(A15)*5.015 / 1024.0)*11.132; //Lee el voltaje del
pin A13 y realiza las operaciones para convertirlo a voltios
PorVolt = (voltaje*100)/ BateriaMax; //Calcula el porcentaje y muestra un
nivel de carga en función del resultado obtenido
if (PorVolt<=75) { nivel=0; digitalWrite(zumbador,HIGH);
} //Cuando la bateria está baja se activa el zumbador
else if (PorVolt>75 && PorVolt<=80) { nivel=1; }
else if (PorVolt>80 && PorVolt<=85) { nivel=2; }
else if (PorVolt>85 && PorVolt<=90) { nivel=3; }
else if (PorVolt>90 && PorVolt<=95) { nivel=4; }
else if (PorVolt>95) { nivel=5; }
Serial.print("Voltaje de batería: ");Serial.println(voltaje); //Imprime
el voltaje de la batería
}
}
void MedirVelocidad(){
sensors_event_t event; //Medir aceleraciones
accel.getEvent(&event);
//Sumadas o restadas los valores de offset para poner a cero la medición en
estático
Serial.print("X: "); Serial.print(event.acceleration.x+0.59); Serial.print("
");
Serial.print("Y: "); Serial.print(event.acceleration.y-0.47); Serial.print("
");
Serial.print("Z: "); Serial.print(event.acceleration.z); Serial.print(" ");
Serial.println("m/s^2 ");
a1=event.acceleration.x+0.59;
Serial.print("a1 1 : ");Serial.println(a1);
t1=millis()/1000.00;
if (abs(a1)>0.05){
v=v+0.5*a1*(t1-t0);

```



```

}else{
a1=0;
}
Serial.print("t1");Serial.println(t1);
Serial.print("t0 ");Serial.println(t0);
Serial.print("a1 ");Serial.println(a1);
Serial.print("a0 ");Serial.println(a0);
a0=a1;
t0=t1;
Serial.print("Velocidad: "); Serial.print(v);Serial.println(" m/s");
Log.print("Velocidad: "); Log.print(v);Log.println(" m/s");
Log.print("X: "); Log.print(event.acceleration.x+0.59); Log.print(" ");
Log.print("Y: "); Log.print(event.acceleration.y-0.47); Log.print(" ");
Log.print("Z: "); Log.print(event.acceleration.z); Log.print(" ");Log.
println("m/s^2 ");
}
void GirarServo(int angulo){
angulo = constrain(angulo, 55, 125); //restringimos el valor de 55 a 125
ServoTimon.write(angulo); //Escritura del valor del ángulo en la salida del servo
Serial.println("Giro a la izquierda");
Log.println("Giro a la izquierda");
}
void TimonRecto(){
angulo = 90;
angulo = constrain(angulo, 55, 155); //restringimos el valor de 45 a 135
ServoTimon.write(angulo); //Escritura del valor del ángulo en la salida del servo
Serial.println("Enderezar timón");
Log.println("Enderezar timón");
}
void GirarIzqSuave(){
angulo = 105;
angulo = constrain(angulo, 55, 125); //restringimos el valor de 45 a 135
ServoTimon.write(angulo); //Escritura del valor del ángulo en la salida del servo
Serial.println("Giro suave a la izquierda");
Log.println("Giro suave a la izquierda");
}
void GirarDchaSuave(){
angulo = 75;
angulo = constrain(angulo, 55, 125); //restringimos el valor de 45 a 135
ServoTimon.write(angulo); //Escritura del valor del ángulo en la salida del servo
Serial.println("Giro suave a la derecha");
Log.println("Giro suave a la derecha");
}
void GirarIzq(){
angulo = 125;
angulo = constrain(angulo, 55, 125); //restringimos el valor de 45 a 135
ServoTimon.write(angulo); //Escritura del valor del ángulo en la salida del servo
Serial.println("Giro a la izquierda");
Log.println("Giro a la izquierda");
}

```

```

void GirarDcha(){
  angulo = 55;
  angulo = constrain(angulo, 55, 125); //restringimos el valor de 45 a 135
  ServoTimon.write(angulo); //Escritura del valor del ángulo en la salida del servo
  Serial.println("Giro a la derecha");
  Log.println("Giro a la derecha");
}

void Brujula(){
  MagnetometerScaled scaled = compass.ReadScaledAxis();
  int MilliGauss_OnThe_XAxis = scaled.XAxis;//
  // Calculate heading when the magnetometer is level, then correct for signs of
  axis.
  float heading = atan2(scaled.YAxis, scaled.XAxis);
  float declinationAngle = 0.089;
  heading += declinationAngle;
  if(heading < 0)
    heading += 2*PI;
  if(heading > 2*PI)
    heading -= 2*PI;
  // Convertir radianes a grados
  float headingDegrees = heading * 180/M_PI;
  Serial.print(headingDegrees);
  Log.print(headingDegrees);
  Serial.println(" Degrees \t");
}

```